ADST

CONFIGURATION MANAGEMENT PLAN

CONTRACT NO. N61339-91-D-0001

Document No. ADST/WDL/TR-92-003023

Date: 1/15/93

Version: 1.3

Prepared by:

LORAL
Western Development Labs

3200 Zanker Road
San Jose, California 95161-9041

Electronic Defense Systems Software Department
Software Engineering Laboratory
Engineering Operation
Loral Western Development Labs

REPORT DOCUMENTATION PAGE

Form approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
15 Jan 1993

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
Advanced Distributed Simulation Technology (ADST) Programs
Configuration Management Plan

5. FUNDING NUMBERS
Contract No. N61339-91-D-0001
CDRL A008

6. COMPILED BY
Bright, Rick; Humber, Karen; Ipsaro, Maria

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Loral WDL
ADST Program Office
3200 Zanker Road
P.O. Box 49041
San Jose, Ca. 95161-9041

8. PERFORMING ORGANIZATION REPORT NUMBER
ADST/WDL/TR-92-003023

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Simulation, Training and Instrumentation Command
Naval Training Systems Center
12350 Research Parkway
Orlando, FL 32826-3275

10. SPONSORING ORGANIZATION REPORT

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (Maximum 200 words)

The ADST Configuration Management Plan describes the CM practices and procedures required for the concurrent, multi-site development and support of the ADST Battlefield Distributed Simulation-Development (BDS-D) programs.

14. SUBJECT TERMS

15. NUMBER OF PAGES
70

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

17. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

17. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

Record of Reviews and History

Rev	Date	Author(s)	Description
1.0	10-02-92	Humber/Bright	
1.1	10-26-92	Humber/Bright	First Draft
1.2	12-01-92	Humber/Bright	Comments from review incorporated
1.3	1-15-93	CM Conference Attendees	Comments incorporated from December 1992 CM Conference.

TABLE OF CONTENTS

List of Figures:

List of Appendices:

1.0 Introduction

This plan establishes the Configuration Management (CM) practices and procedures to be followed in support of the Advanced Distributed Simulation & Training (ADST) contract by Loral Western Development Labs (WDL). The plan describes the CM practices and procedures required for the concurrent, multi-site development and support of the ADST BDS-D (Battlefield Distributed Simulation-Development) programs. This plan addresses the Software Development Facility in San Jose, Orlando, subcontractors working on ADST Delivery Orders (DO's), Ft. Knox and Ft. Rucker.

A contractor managed, Configuration Control Board (CCB) will be established at the Loral WDL, San Jose facility, which will monitor, approve, and control changes to the common ADST product baseline. The centralized CCB will be responsible for adjudicating baseline changes proposed by the various development, operational, and integration functions. This plan describes the configuration change control procedures to be followed by the CCB and identifies the relationship to alternate development and integration sites (i.e., Orlando, Ft. Knox and, Ft. Rucker) and subcontractors working on ADST Delivery Orders (DOs).

Each Delivery Order (DO) organization will apply standard practices and processes to its design/development and integration activity. Systems Engineering will be responsible for the initiation, development, and coordination of all DO's on the ADST program. CM will work closely with software and systems engineering to assure consistency, and integrity of the initial submissions, procedural changes and traceability of software and hardware components. Systems Engineering, along with the Program Engineering Manager will be a liaison with the CCB and will coordinate with software engineering, integration and CM on software deliveries. In order to ensure consistency between sites and DO's, and adherence to the policy and procedures set forth in this plan, CM operations are coordinated with the BDS-D Manager, Site coordinators, and the Loral WDL Program Office.

A key component of the plan will be the identification and establishment of the common ADST software baseline as a reusable system component as well as identifying site unique differences. The control mechanisms described in this plan serve as the foundation of a tailorable standard applied to all ADST BDS-D Delivery Order (DO) efforts in addition to ongoing baseline maintenance.

The LORAL CM system described in this plan is based upon and complies with DI-CMAN-80858, MIL-STD-1456A as tailored to the ADST program. Although the ADST program is not a MIL-STD project, MIL-STD-1465A definitions and terminology are used throughout this document.

1.1 Description of the SIMNET Configuration Item (CI)

ADST is a direct descendent of the DARPA/Army SIMNET program. SIMNET represented a major advance in the Armed Forces ability to generate selective fidelity representations of the real world in low-cost simulators for combined forces training and system evaluation. However, a milestone has now been reached. The proof-of-concept program is complete and the SIMNET program is advancing into a new phase. ADST will continue research and development of additional simulators and functional capabilities, but this research must be directed by user developed requirements. Advancements will be phased into the laboratory testing environment through procedures that also support other on-going test and evaluation processes.

The Mission of ADST includes the test and evaluation of conceptual combat systems; training effectiveness; organization, tactics, and doctrine; and simulation network elements.

The Key mission objectives of the ADST contract are stated below:

- BDS-D will be a key research tool for faster and more successful development of weapons and command and control systems.
- BDS-D will be used in evaluations to define the specific training tasks and methods of assessment for simulator training.
- BDS-D provides an opportunity to view and evaluate the effects of changes in organization, tactics, or doctrine in realistic combat situations.
- BDS-D will be used to support developmental testing and validation to expand the simulation network itself.

Configuration management is the cornerstone of a disciplined process. Existing simulation network elements must be functionally documented and placed under the configuration management process. Most critically, supplemental documentation must be prepared for functional and performance baselines of existing equipment, so that test proponents know the capabilities when planning investigations of critical test issues.

ADST is at the center of a rapidly changing world of simulation and simulator development. To capture ADST's role in that world requires the creation and maintenance of an ADST system description under a formal configuration management program.

1.2 Program Phasing and Milestones

ADST maintains a detailed software development schedule for each Delivery Order/CSCI's. The schedule is based on the overall program schedule as well as on a comprehensive software development methodology, based on Company Standards and Procedures, and as streamlined to adapt to an Operations and Maintenance program like ADST. The determination of specific major milestones are influenced primarily by Contract requirements specified in each Delivery Order.

Appendix A and B provide current status of the program at the time of preparation of this plan. Appendix B provides a CM activity chart depicting current milestones. Both plans are dynamic and will be updated as required.

1.3 Special Features

The ADST program contains a number of peculiarities that require special attention and understanding in order to provide the Configuration Management services needed to establish a firm foundation for the BDS-D system baseline. Issues described in the following paragraphs that make the ADST program unique are as follows:

- existence of a research and development product baseline,
- delivery orders that represent ECP like changes to the baseline,
- requirements for delivery order funding of CM,
- parallel development efforts with no baseline control,
- multi-site development efforts, and
- multiple H/W and S/W configurations.

The ADST Configuration Management (CM) funding was provided 18 months after the program started under the BDS-D Software Support Contract Change Proposal (SWCCP). This effort was designed to bring a poorly documented software baseline developed as a research and development effort, running at 3 different sites, on poorly documented hardware configurations, under baseline control. The objectives of the baseline control activities to be provided are as follows:

- inventory the software,
- identify the various software configurations,
- identify the various hardware configurations,
- establish procedures for change control,
- establish procedures for release control,
- establish build procedures for each software configuration,
- provide version description documents for the various builds,
- maintain records, audit trails.
- support multi site development, and
- provide a development environment for test and integration.

The funding profile for the CM was reduced substantially from what was originally proposed. In order to maintain a subsistence level of support for configuration management it was deemed necessary to have each delivery order fund it's own CM in addition to the sustaining effort. This has the advantage of only funding additional help when the work load increases above what the sustaining effort can support due to delivery order work, and then reducing the staff when the work load decreases. Problem and tracking is part of the sustaining effort but fixes are outside the scope of the CM effort. At the same time the sustaining effort maintains the knowledge of procedures, builds, and configurations required to maintain continuity for the program.

Another problem being addressed by the SWCCP is that of development that took place when no standards or procedures were in place. In fact, there was apparently no CM control beyond the knowledge in the heads of engineers concerning the structure and organization of software for the various simulators. This has resulted in "builds" that are incomplete, undocumented, software that is out of date, utilizing procedures that were never adequately finished, to perform the compilation and links. Slowly, configuration by configuration, engineering expertise has been brought to bear to resolve these problems. Missing files have been identified and obtained, build files corrected, and procedures put in place to address the turnover and release of software. This time consuming process requires reverse engineering of the software, and analysis of the interfaces.

Another element peculiar to the ADST program is its multi-site development aspect. Utilizing electronic networks and by subcontracting software development, concurrent development is able to take place on a delivery order. This requires careful coordination to ensure that software changes to the same module are identified and merged prior to testing and integration. A tracking mechanism for software Check-in and Check-out has been developed. A review cycle will be utilized to ensure the integrity of the baseline for those modules where concurrent development is taking place.

Finally, ADST has to deal with multiple software and hardware configurations. One delivery order may make changes to a simulator, by changing its hardware and/or software, to run a series of experiments. In some cases the changes become permanent, and in other cases they don't. Until the SWCCP was funded there was no way of tracking the software builds to a required hardware configuration. That has changed with the institution of CM procedures. The CM organization will be able to archive and retrieve old builds, clearly identify the hardware configuration a software build runs on, and provide a version description document of the code required to reconstruct an experiment.

In summary, the ADST program represents a large, complex body of hardware and software, operational at several different sites across the country. To successfully implement a viable and credible configuration management program requires the cooperation and participation of government, subcontractors, and Loral personnel. This plan is written to focus and present a process to correct the inherited baseline and continue configuration managment practices throughout the lifecycle of the program.

2.0 Organization

The ADST Software Support team and the ADST Program Office will be supported by a Software Configuration Management team. Technical authority is provided by the ADST Program Engineering Office. Figure 2-1, Software Support Organization, illustrates the organizational relationships of the Software Configuration Management team to the ADST Program Office and DO Managers.

Initial CM objectives require the identification of the SIMNET 6.6.1 software baseline and documentation required to establish the various configuration baselines. Loral will establish software product baselines against the SIMNET 6.6.1 release and manage continued releases as they are completed on existing and new delivery orders. Configuration reviews, audits, and the formal operation of the Configuration Control Board (CCB) will ensure proper application of configuration control and establish the basis for status accounting.

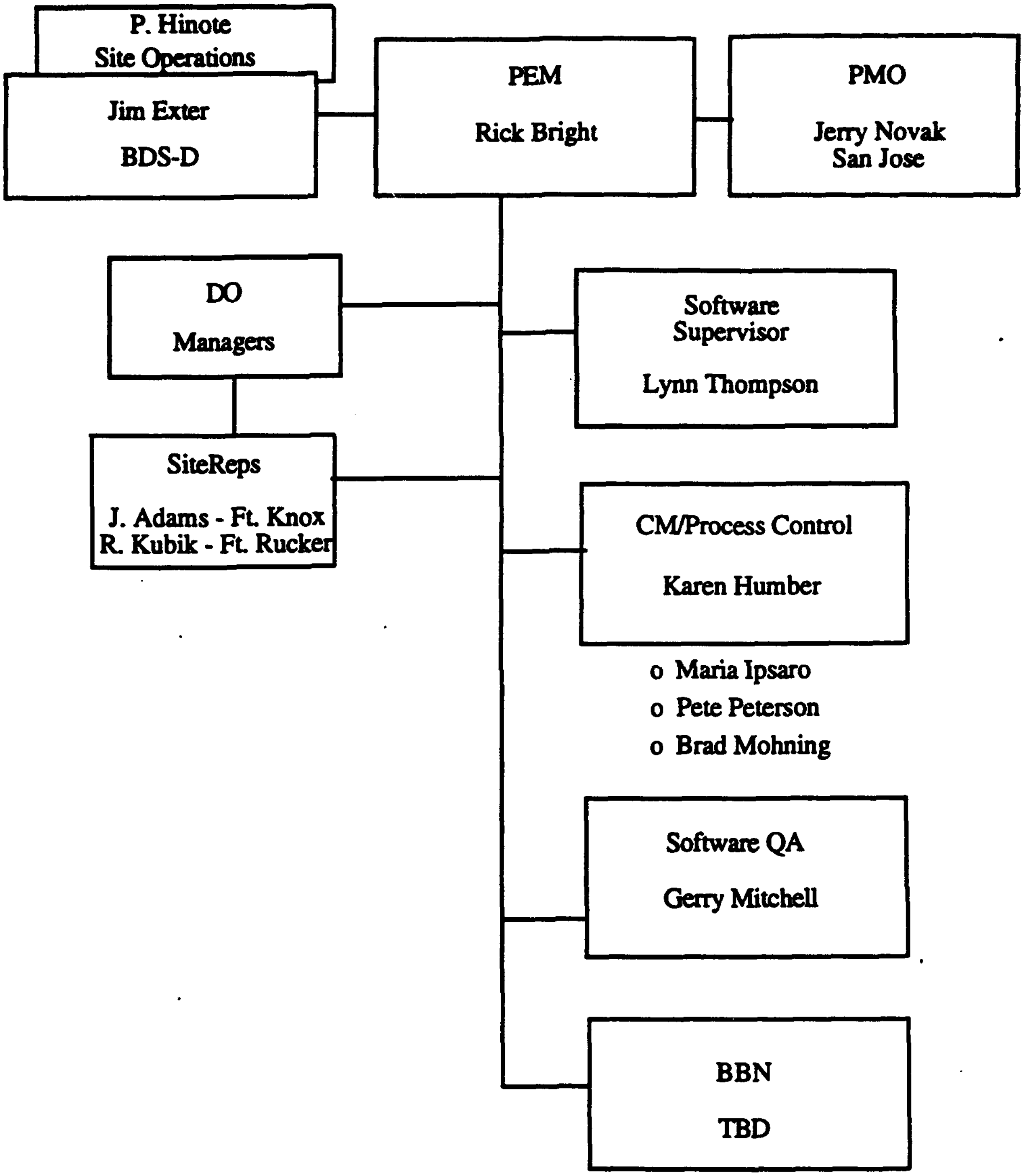


Figure 2-1. Software Support Organization

2.1 Structure

2.1.1 Structure (CM)

LORAL WDL responsibilities for Configuration Management are assigned to the ADST Software Process Control Section. LORAL will apply CM directives and operating procedures that have been established at the Division level, and are responsive to Program requirements. LORAL will provideCM as described herein, and will apply the standards of this plan to subcontractors as appropriate.

Within CM, a program planning and support function will establish the tailoring of routine internal procedures to the specific program requirements. The established CM team will provide active program support for coordinating, monitoring and implementing the activities that result in appropriate configuration identification, control, and status accounting. Detailed responsibilities will include the following:

a. Assurance that the configuration of all deliverable items are fully identified.

b. Perform Configuration Management Procedures IAW ADST Software Standard Operating Procedures (Appendix C).

c. Establishment and maintenance of the Software Support Library (SSL).

d. Establishment of procedures for Software build loads and installation.

e. Traceability of changes to field patches, and maintain accurate software configuration data for each site.

f. Assurance that configuration data of all delivered and in-process equipment is maintained.

g. Preparation, maintenance and distribution of configuration status accounting reports.

h. Coordination of, and preparation for, CCB activities and CM audits.

i. Exercise of change control resulting from Software Problem/Change Reports (SP/CR's) and Engineering Change Proposals (ECP's).

j. Maintain SP/CR status for all SP/CR's.

k. Baseline documentation maintenance.

l. Maintenance of Hardware and Software Inventory.

m. Development of ADST CM Software Standard Operating Procedures.

2.1.2 Configuration Control Boards

2.1.2.2 Configuration Control Board (CCB)

The Loral, San Jose, WDL Configuration Control Board (CCB) is designated as the Central CCB (i.e. single control point) for all changes to the established BDS-D software product baselines. In order to ensure consistency between the site development changes and the central software baseline, all CM related operations require communication and coordination in cooperation between the San Jose CM staff, DO Managers, and Site representatives. The Sites will apprise the CCB of any possible or proposed changes to the product baseline. The respective DO Manager and Site Representative will review and approve site generated SP/CR's. Configuration reviews and the CCB will ensure proper application of configuration control and establish the basis for status accounting. The CCB will provide active program support and establish a team for coordinating, monitoring, and implementing change control. The CCB will interact with the LORAL Orlando Program Engineering Office, Site representatives, and with the customer to adjudicate proposed changes to the baselines. Role and Responsibilities for the CCB are defined in the ADST Standard Operating Procedure, SSOP-002, in Appendix C.

2.2 Authority and Responsibility

An overview of the Loral WDL ADST organizational structure as it relates to Configuration Management is presented in the context of the ADST Program Organizational structure was shown earlier in Figure 2-1. The ADST program is organized under a Program Management Office (PMO) that directs the total effort. This office employs the resources within the LORAL WDL Electronic Defense Systems (EDS) Software Department through a formal work tasking and authorization procedures. Control of the ADST activities is exercised at the highest level by the ADST Program Manager, supported by the ADST Program Engineering Manager (PEM) who has direct responsibility for all engineering activities. The ADST Software Engineering Supervisor, reporting to the ADST PEM, has full responsibility for all aspects of software development from requirements analysis, code check-in, through software releases. He is supported by the EDS Software Process Control Section for Software Configuration Management (SCM), Software Systems Administration, methods, standards, and metrics.

The EDS Software Process Control Supervisor reports progress and status to both the Software Engineering Supervisor and to the ADST Program Engineering Manager (PEM), thus providing an independent reporting path. Software quality evaluations are performed by the Software Quality Assurance (SQA) organization (which is independent from the ADST Program). This independent quality assurance capability ensures that all processes are conducted, and that all products are developed, in accordance with contract requirements and that Configuration Management is conducted in accordance to this CM Plan. The Baseline Configuration is maintained by the Software Configuration Management (SCM) Analysts and Software Systems Administrator who are members of the EDS Software Process Control (SPC) Group.

The CCB will convene on an as needed basis and will be composed of members from the Program Office, DO Managers, Software Engineering, Systems Engineering, Configuration Management, Test, Software Quality Assurance, Sites and Government (STRICOM) representation. Actions of the LORAL CCB will include verifying compliance with contractual requirements and ensuring the identification, evaluation and consideration of the technical reasons for changes(s). Members of the CCB will advise the Chairman (ADST PMO or designee), regarding the impact that proposed changes will have in the areas of costs, production, and documentation support.

2.3 Policy Directives

An independent CM organization with defined responsibilities, policies, and procedures that comply with ADST program requirements has been established in compliance with policy directives. Process control and configuration management practices have been tailored for the ADST program using Company directives and standards from the LORAL WDL Software Productivity Laboratory and the Software Engineering Process Group.

2.4 Reference Documents

Government Documents:

MIL-STD-1456A	Configuration Management Plan

Non-Government Documents:

ADST SSOP's	ADST Software Standard Operating Procedures - CM Procedures
ADST LWDL	Documenation Plan for the ADST Program
ADST LWDL	Software Quality Program Plan (In process)
ADST LWDL	Software Development Plan (In process)
ADST WDL/TR--92-03014-QTR1YR2	System Definition Document
Software Productivity Lab (SPL) Documents	
Vol. 1	Corporate Standards Methods
Vol. 2	Software Engineering Practices

3.0 Baseline Identification

The ADST team approach to managing the software life cycle will be based on establishing a product baseline. The Product Baseline (initially SIMENT Version 6.6.1) will be used as the starting baseline for subsequent design development and/or DO's. Changes to the product baseline will be established to define a formal departure point for control of future revisions, or deliveries. In the event that a new simulator is developed, configuration identification will be established in the form of technical specifications for the system, the system segments, and each Configuration Items (CI/CSCI). Hence, Functional Configuration Identification (FCI) will be established against performance oriented requirements for the segment design and performance.

The ADST project involves the development and maintenance of many complex software components which execute on several hardware platforms. The project software development is not for the purpose of achieving one delivery, rather for the purpose of achieving multiple releases with potential increasing capabilities for a particular configuration.

The elements below comprise the Engineering groundwork required to support a DO Software release process. The elements required for a DO package may vary based on requested sponsorship and will be funded with each DO. The release kit, including the code is for general release unless specified by the customer. The information required forms the basis for the "Deliver Order Release Kit" (SSOP-0009):

- Delivery of a Version Description Document depicting the contents of the build release. Including associated open and closed SP/CR's. The VDD will be distributed on floppy disk along with a hardcopy.
- Provide required executable code, object code, data files, tables and parameter files to bring up a system. Include a read.me file with appropriate header information to identify tapes for loading software.
- Coldstart procedures.
- Identification of the hardware configuration required by a softwware release.
- Installation procedures.
- Build and distribution instructions.
- Identification of known problems.
- Regression test results.
- Reference to the Acceptance Test Plan for used recertify that a configuration has been restored to its "baseline" configuration.
- English narrative synopsis of functionality of the released system.
- List of documents applicable to the configuration.
- Release Notes, if appropriate, depicting notes/differences/compatibility since the last release.

- Provide tools to support any necessary on-site regression testing.

- Provide POC (name/phone number) for each release.

- User Guides (as funded by a DO).

Configuration Identification and control is maintained throughout all software development phases. The following products, as required by the sponsoring organziation will be maintained by the ADST Software Configuration Manager:

- Software Development Plan (SDP)
- Configuration Management Plan (CMP)
- Software Requirements Specifications (SRS)
- Interface Requirements Specification (IRS)
- Software Product Specifications (SPS)
- Software Test Plans (STP)
- Software Test Descriptions (STD)
- Software Development Files (SDF)
- Software Design Documents (SDD)
- Interface Design Documents (IDD)
- Version Description Documents (VDD)
- Software Maintenance Manuals (SMM)

3.1 Product Baseline

The product baseline will be established for the system by the CI software product specification after successful completion of an informal LORAL WDL Functional Configuration Audit (FCA) and Physical Configuration Audit (PCA) for each CSCI. The product baseline includes the identification of evolving source code and software build loads (executable code).

3.1.1 Specifications

As applicable for a Delivery Order, the product baseline will be documented in the Software Product Specifications and Interface Specifications. In the case of SIMNET 6.6.1, however, the product baseline is defined without any underlying CSCI structure.

3.1.2 Drawings and Associated Lists

Drawings are to be prepared to WDL DPM practices and on WDL drawing format. All drawings will be prepared to Level I requirements unless otherwise stated in the contract/delivery order. Drawings originating within WDL San Jose and at Orlando and any other locations will be submitted to the ADST PMO, WDL San Jose CCB review/release to the Engineering Data Management System (EDMS) database. Custodial release is authorized as the cost effective drawing control approach to this prototype development research and analysis type system. Custodial release is described in the Documentation Plan for the ADST Program available from CM.

4.0 Configuration Change Control

The purpose of change control is to prevent incorrect or unapproved changes from taking place while expediting the approval and implementation of formally approved changes. These changes are grouped into the following categories:

a. Software Problem Fixes

b. Hardware Problems

c. Documentation

d. Firmware

e. Delivery Order (DO) software updates

In addition to change decision making, change control includes the equally important functions of setting change priorities (emergency, urgent, or routine) and of assuring that necessary instructions are issued promptly for approved change. Internal change control will be applied to the entire software product throughout the development, maintenance, and informal test phase.

Change control of the product baseline is tracked through the use of the Software Problem/Change Report (SP/CR) form. The SP/CR is the vehicle used internally to report a known or suspected anomaly in the developmental configuration for software, hardware, firmware, or documentation. The SP/CR procedures and guidelines are described in ADST SSOP's-0003 and 0006.

4.1 Responsibilities

The ADST CCB process is to assess, review, and evaluate the necessity for proposed changes to approved product baselines. Release of new design(s), and changes to approved baselines require approval of the Loral WDL CCB. The CCB is chaired by the Loral ADST Program Manager in San Jose, and is composed of senior members of the various support organizations including Software Configuration Management. The CCB will meet on an as needed basis, the frequency and time of meetings to be determined by the chairperson. Detailed membership and roles and responsibilities for the CCB are described in SSOP-2, "CCB Charter".

The sites are delegated the authority to approve site implemented SP/CR's. Each site will have the responsibility to maintain the developed products in site development directories. Once functional, the products will be transmitted back to the San Jose CM team, reviewed at the Central CCB and incorporated into the SSL baseline repository. Procedures for site turnover of code and documentation are outlined in SSOP-0007. The Central CCB will mediate conflicts between sites due to changes (hardware, firmware, and software product configuration items) and analyze proposed DO updates to a software product baseline. Release of specifications, new design(s), and changes to approved baselines require approval of the CCB.

Configuration Management will control, document, and distribute all software releases to the test and target environment in the Loral Software Development Facility (SDF), as well as to the operational sites. It will be configuration management's responsibility to create software construction procedures which are repeatable for any revision that been introduced into the Software Support Library. CM will document the exact configuration of all hardware and software used during test and integration. A comprehensive list of all hardware and software configurations in the Loral SDF are maintained in the ADST System Definition Document (SDD).

4.2 Procedures

4.2.1 Software Configuration Control Process

The primary reporting mechanism for changes to the BDS-D/DO system configurations is via the Software Problem/Change Report. The SP/CR is used by the sites and the CCB to resolve design and document issues to the CM controlled software baseline. An SP/CR status report, which serves as the agenda, is prepared and distributed by Configuration Management in a timely manner to support the functions of the Board. Since the CCB representatives may be located at various facilities, technical interfacing will be accomplished via the conference phone.

The following is the detailed process flow(s) associated with each SP/CR:

a. Accept the SP/CR. Evaluate cost, schedule and/or project performance. Evaluate whether or not the requested change is a common software product change or a site unique change. Review responsibility to work; task software engineer to implement the change.

b. Defer the SP/CR for further study. If the change is out-of-scope of the LSE contract, direct Systems Engineering to prepare documentation for subsequent approval.

c. Disapprove SP/CR. Notify originator of documented disapproval; close SP/CR.

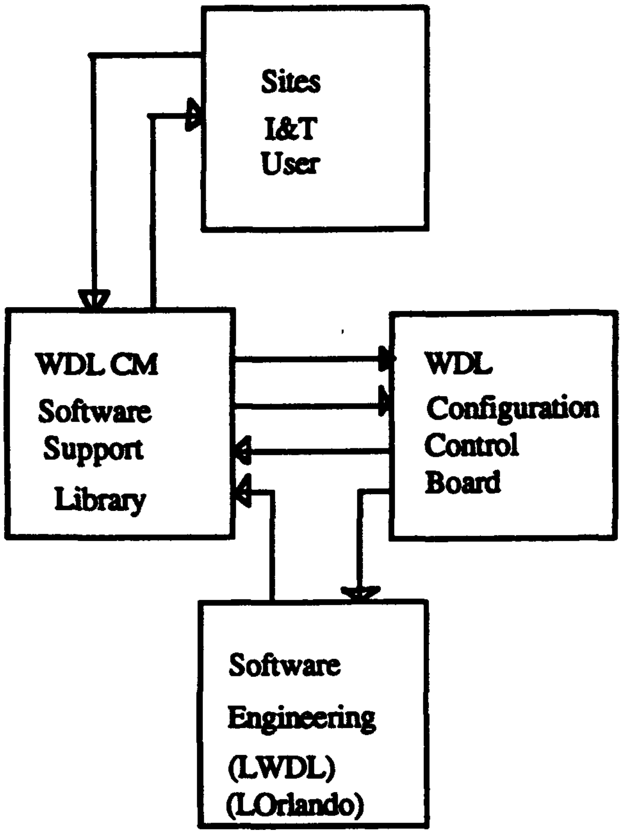
d. Identifying problems and request funding for out-of-scope issues.

Status on all of the above actions from the CCB will be provided to the ADST Program Office and the Delivery Order Mangers.

4.2.2 Software Problem/Change Report (SP/CR) Processing

The SP/CR provides the means for reporting errors in development, test, and operational software and/or hardware. An SP/CR is completed by any user or developer who discovers a system anomaly, or, desires a modification in order to improve system operational efficiency. The SP/CR provides a formal means for recording all software and hardware faults and modification requests. Each SP/CR will be submitted to the CCB or respective site representative/DO Manager for analysis. SP/CR's which are out of scope, such as beyond the normal schedule, scope of current activities, budget or enhancements/improvements will be forwarded to the LORAL WDL Program office for resolution. Figure 4-1, Change Process, provides an overview of the ADST software/change correction process.

After software, hardware, and documentation are placed under configuration control, all modifications to these products require an SP/CR. The SP/CR assignee is authorized to resolve a problem only by being designated by an approved SP/CR. The SP/CR resolution must be verified by peer review before recommending closure of the SP/CR. The on-line history in the code must be updated along with associated documentation , including build requirements to reflect specific changes. When completed, the assignee then coordinates the recommended closure of the SP/CR with the Software Configuration Manager. No SP/CR is closed until successful completion of verification and test with CCB concurrence or the Site Representative/DO Manager.



1. Sites discover problem, generate SP/CR reports, including analysis.

2. SSL using 4D logs problems and submits to the CCB.

3. Site/CCB analyzes problem
a. Within scope of LSE?
b. Relevant to current DO?
c. Out of scope?
If a or b

4. Engineering fixes problem and conducts approprite tests. Submits fixes to CM/SSL.

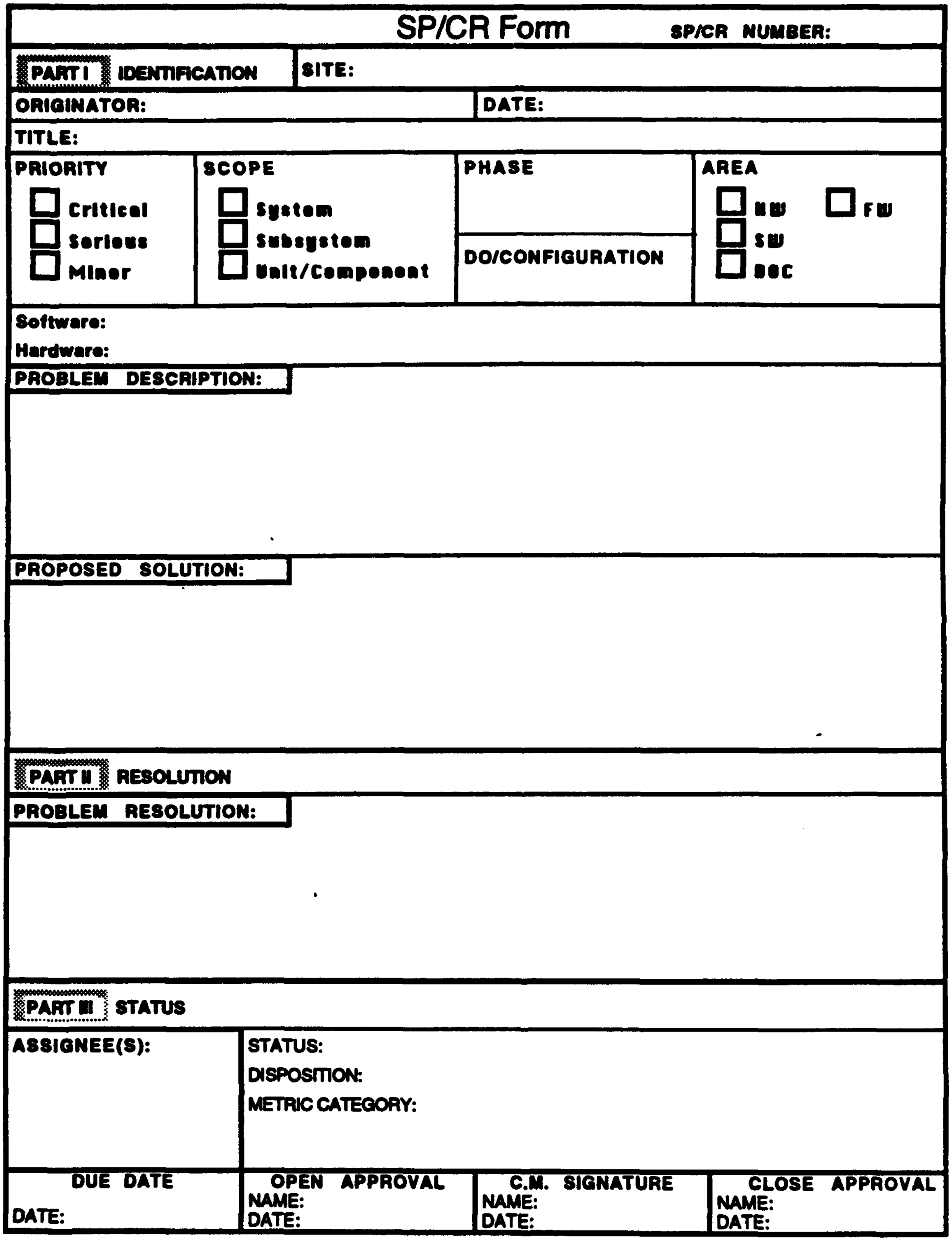
5. CM/SSL logs fixes and submits to CCB.

6. Site/CCB reviews fixes and approves update to the baseline.

7. CM/SSL updates the baseline and releases it to the I&T/Site.

Figure 4-1. Change Process

The ADST SP/CR system is a multi-user, multi-site Software Problem Reporting application using the 4th Dimension Data Base Management System (DBMS). The SP/CR system is located at LORAL WDL and can be accessible from each site with the installation of the 4D Runtime License. The SP/CR system is a menu-driven tool with data accessibility divided into four groups; the submitter, the approver, the assignee, and configuration management. An ADST SP/CR User's Manual will be available on-line. The procedures are documented in SSOP-0003. Figure 4-2 is a sample of the SP/CR form. SSOP 0003 shows samples of the SP/CR data base menus.



SP/CR Form SP/CR NUMBER:

PART I IDENTIFICATION SITE:

ORIGINATOR: DATE:

TITLE:

PRIORITY □ Critical □ Serious □ Minor	SCOPE □ System □ Subsystem □ Unit/Component	PHASE DO/CONFIGURATION	AREA □ HW □ FW □ SW □ DOC

Software:
Hardware:

PROBLEM DESCRIPTION:

PROPOSED SOLUTION:

PART II RESOLUTION

PROBLEM RESOLUTION:

PART III STATUS

ASSIGNEE(S):

STATUS:
DISPOSITION:
METRIC CATEGORY:

DUE DATE DATE:	OPEN APPROVAL NAME: DATE:	C.M. SIGNATURE NAME: DATE:	CLOSE APPROVAL NAME: DATE:

Figure 4.2. SP/CR Form

4.2.3 Software Change Control

The control process for each product or Delivery Order (DO) under development adheres to a set of activities that is accomplished in a particular order to conform to the building block approach of SP/CR validation and verification. The CM tools and directory structure provide CM with the ability to uniquely identify and build a DO product baseline. In this process there are related phases that provide the ability to incrementally build on a foundation of a tested, controlled, and validated baseline. There is one CM library area that is managed and maintained by the CM team. This domain is referred to as the Software Support Library, (SSL) described in further detail in paragraph 4.2.5 below. An identical structure is created from the CM area for developers to perform upgrades. Here developers code and build their changes. After successful development / unit test on the target machine, the change is turned over to CM for control and incorporation into the CM baseline via an approved SP/CR. Figure 4-3 provides a graphical view of the CM structure.

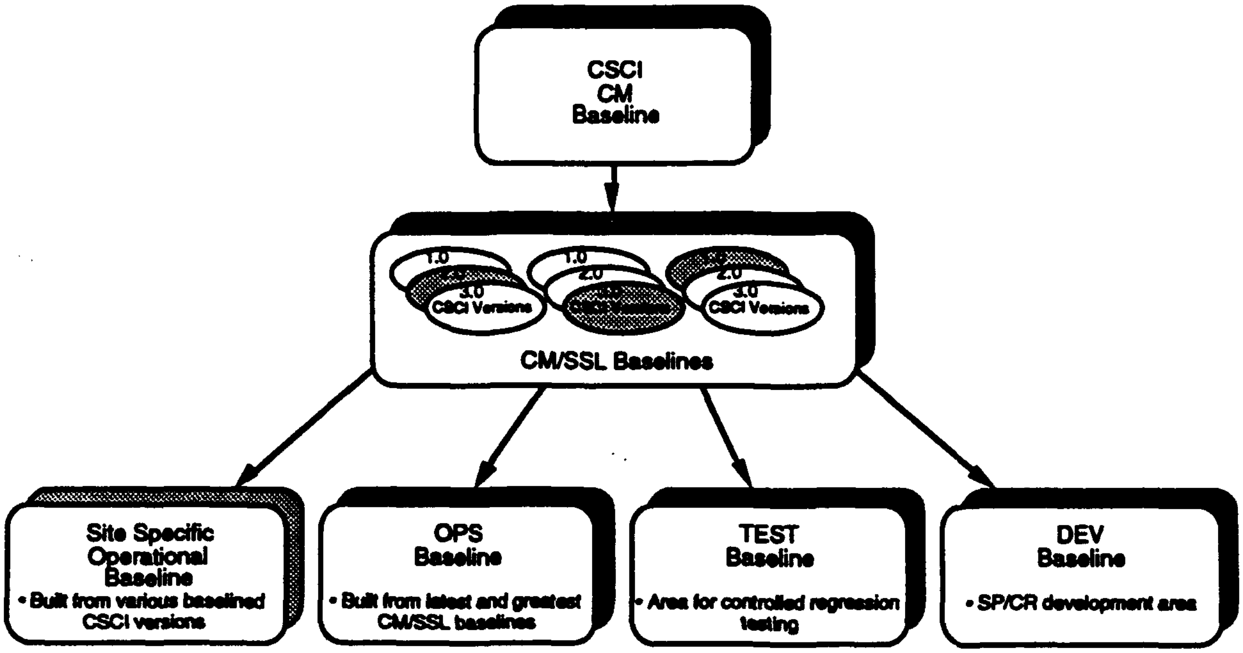


Figure 4-3. SSL Structure

The CM/SSL Baseline structure is a controlled environment, which provides protection from unauthorized software changes. It contains modules which have been identified by software engineers for control and are managed by the Configuration Management team under the Revision Control System (RCS) or other CM software packages as applicable. These modules have been built (compiled and linked) to produce an application build load or Delivery Order (DO). A build is defined as the compilation and linking of source code into the program executable images that run on the target simulator. Included as part of the build are any data files or databases required to initialize and/or support the execution of the programs in the simulators. Each build load is flagged via RCS with a revision number associated with the particular DO. Changes to software modules will be identified by an SP/CR number in the header / prologue block of the source. The SP/CR will remain open until the change has been validated through regression testing in a closely monitored integration environment.

The development, test, and operational software baselines are maintained in directory structures on the target machines under the SSL that are closely monitored by CM. The CM Baseline directories contain a snapshot of the controlled baselines, which can be used to recreate a given environment. The Operational Baseline directory contains software which has been derived from some baseline, upgraded, tested, controled, and labeled "operational". This directory and all subdirectories are under CM control by use of operating system protections. The Test Baseline directory contains a CM built baseline, which is not yet operational. Formal testing of software "captured" from developers is performed in this area. When this testing process has determined that the software is working properly, it is moved into the Operational baseline directory by CM.

The Development Baseline area is where build loads are downloaded and distributed to the designated development configurations in the SDF for SP/CR incorporation and further development and unit testing.

4.2.3.1 Development Phase Change Processing

The software development and build process is divided into three areas:

a. SP/CR Initiated/Code Controlled

b. Regression Test/Close SP/CR

c. Tested Baseline

The development and build process will be coordinated with all sites at the LORAL WDL Software Development Facility. For all sites, the SP/CR initiates the introduction of new source into the baseline or identifies a need for change to source already under CM control. The build cycle starts when an SP/CR is submitted for open approval by an originator and ends when the source files are turned over to CM and placed under control. Procedures for checking code in and out from the CM SSL are identified in ADST SSOPs 0005 and 0007.

Once a completed SP/CR has been generated, the engineer ensures that it contains all pertinent data including:

a. description and proposed solution of change,

b. priority of change

c. source files affected by change, and

d. identification of obvious dependencies (other affected source files) with change.

As a result of formalized reporting utilizing the SP/CR process, similar (related/dependent) errors may be coordinated and scheduled with an alternate site at the same time. It is important that a software change does not create problems in other software source files.

If an SP/CR is error free, the source is controlled and all SP/CR dependencies are identified. In this way the configuration manager maintains the internal consistency and correctness of the baseline.

4.2.3.2 Integration and Test/Change Processing

After a package of software is completed it must be tested. In coordination with the site engineer, CM will construct the build configuration to support testing in the SDF. The goals of CM during Unit and Formal testing are to minimize divergence between baselines; allow the software engineers to find and fix bugs quickly; and to build a project baseline containing validated code. To provide the software engineers at all sites with a completely stable test environment, a read only copy of the current built baseline will be made available to the site responsible engineer. This enables the engineer to make a copy of the stable CM controlled baseline, and compile and link within their workspace, without affecting other development performed concurrently. The engineer is able to confidently test against the most current baseline before turning the completed software over to CM. During the Integration and Test Phase the Integration and Test Manager provides a selected list of SP/CRs to be included in the build. All SP/CRs included on this list have been turned over and revalidated by CM, approved by the Site/CCB, and a dependency analysis has been performed against the listed SP/CRs and the controlled baseline. CM then copies the files associated with the SP/CRs indicated in the selected list, performs the build in the appropriate directory tree, and distributes the software to the appropriate SDF Development and Integration and Test machines. Distribution Procedures are documented in SSOP 0010. Figure 4-4 depicts the process flow.

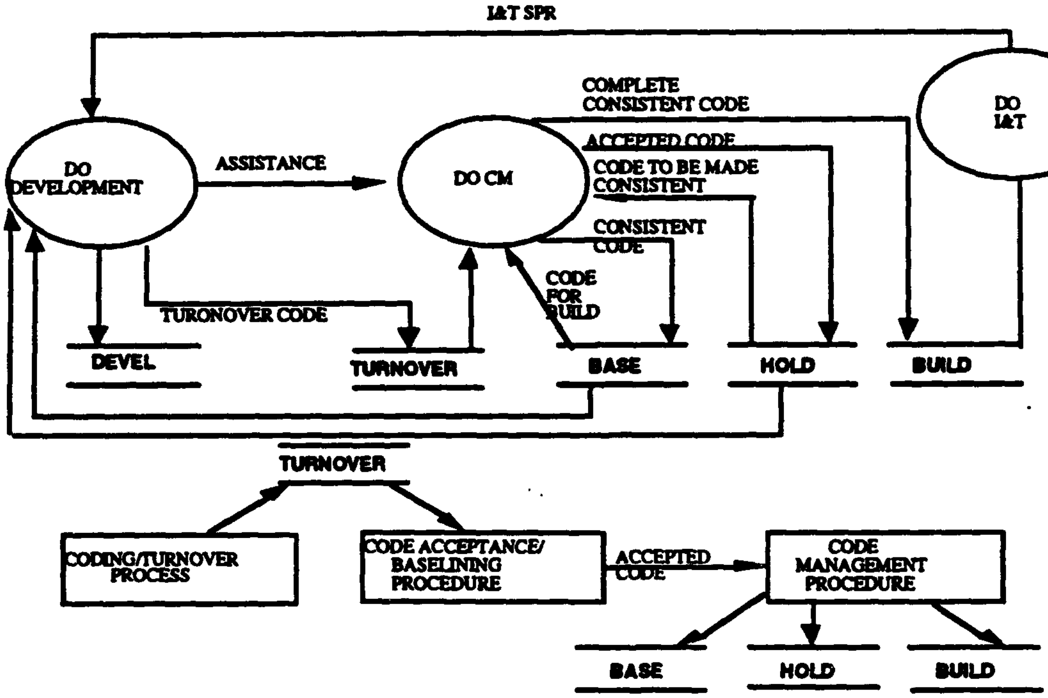


Figure 4-4. Development/CM/I&T Process Cycle

4.2.3.4 Site Integration Change Processing

Any changes to a build during Site Integration will be made in development directories in a Site Integration Development environment. An SP/CR will be generated for all required changes. The code will be modified, tested, and the corresponding SP/CR submitted to the Site Representative and DO Manager for approval. The CCB will review the SP/CR and those SP/CRs receiving approval from site will be incorporated into the baseline libraries. The board will have representation from all software efforts under the ADST umbrella and thus the impact of code modifications can be assessed at this time. If the code under review will further impact other efforts, it will be determined whether this code will go into the common set of baselined code, or other specific baselines. Change control for Field Support is identified in SSOP-0013.

4.2.3.5 Merging Software Baselines

Development of changes may take place at different sites. During this multi-site development effort there will be divergent paths that need to be merged together at discrete points. These discrete points or software revisions will be evaluated with regard to criticality to the user and developer and revision priorities will be established by the joint agreement between the Site and the CCB. The merge process is required when the official baseline has been changed by another development effort. All changes developed using the original SSL CM baseline as the starting point are identified and are integrated into the CM baseline at the SSL to produce a new combined version. A module that has been identified as being modified at more than one site will be compared, and if necessary, a responsible software engineer will be required to perform the merge. Prior to controlling the source, CM will recompile/relink all source and perform an impact assessment against the controlled baseline prior to CCB review.

4.2.4 CM Software Release Naming Conventions

Standard naming conventions are to be used when establishing new and revised BDS-D software versions. The accepted convention will be used for renaming existing 6.6.1 baseline versions and for new baseline versions.

Example

BDS-D RWA Build Version 1.1.0
1. = Major Revision, implies functionality change
1.1 = Minor Revision, implies improvement
1.1.0 = Correction Revision, implies bug fixes

4.2.5 Software Support Library (SSL)

The ADST Software Support Library (SSL) will be established at LWDL San Jose, as an access-controlled environment for controlling the storage, handling, and release of software and related documentation to all supported BDS-D sites. Major SSL functions include the following:

a. Checkin , validate, and place under CM control source code and all related libraries delivered via acquisitions from subcontractors.

b. Generate software masters and copies.

c. Generate source/executable object programs on protected media.

d. Provide protection from unauthorized program changes.

e. Receive and incorporate approved source code changes.

f. Ensure that changes are incorporated in related software documentation.

g. Maintain SP/CR databases.

h. Receive and store test data and programs for verification testing.

i. Provide source and test data for development and testing of software changes.

j. Provide centralized storage of reference, baselined, and deliverable documents.

k. Maintain a tape backup library for all vendor software used on the development and target machines.

The CM tools and directory structure provide the CM group with the ability to uniquely identify and build a specific baseline. There is one CM library area that is managed and maintained by Software Configuration Management (SCM). Source files in this library are marked with a unique version control number and each changed module is associated with an SP/CR identifier in the header of the source. The SP/CR remains open, however, until the fix has later passed regression testing. Modules that check out successfully are put under CM control. A copy of the software identified by the SP/CR is placed in a holding area and waits until called upon by the test team to be included in a build. Procedures for this process are identified in SSOP #0007. When the SP/CR identified software is selected, a copy from the controlled area is moved to a build area and built by CM. If the built software fails during test it is reassigned to the cognizant engineer for rework, which will require a CM revalidation, recompile and relink.

The storage and maintenance of the ADST software products shall be centralized in an automated SSL environment which is the responsibility of the Configuration Manager. This on-line library resides on a SUN SPARCserver. The software configuration management, library control, and system build facilities are performed under the automated techniques of the SUN/UNIX RCS (Revision Control System) code management system and the Make capability (an automated build tool).

The SSL provides unique identification of all baselines, programs, documents, tools and procedures entered into the library. CM prepares, maintains, and periodically disseminates an up-to-date index of all items in the library. CM prepares, logs, and maintains Software Problem/Change Reports in the SSL and provides approved users with the latest CSCI baselines, terrain databases, supporting documentation, and prepares and submits software Version Description Documents (VDDs) for each formal release.

CM has the primary function and responsibility for maintenance and control of all baselines and on-line configuration items. Only the CM staff may access controlled/baselined library information and/or files.

A CM library hierarchy has been established and includes separate RCS libraries for the specific baselines. Included in each baseline are source code, test plans, and documentation. This library hierarchy mirrors each vehicle and simulation subsystem development directory structure, thus facilitating the transfer of data while maintaining consistency of identification and design. Figure 4-5 is a sample of a high level overview of the BDS-D SSL directory structure.

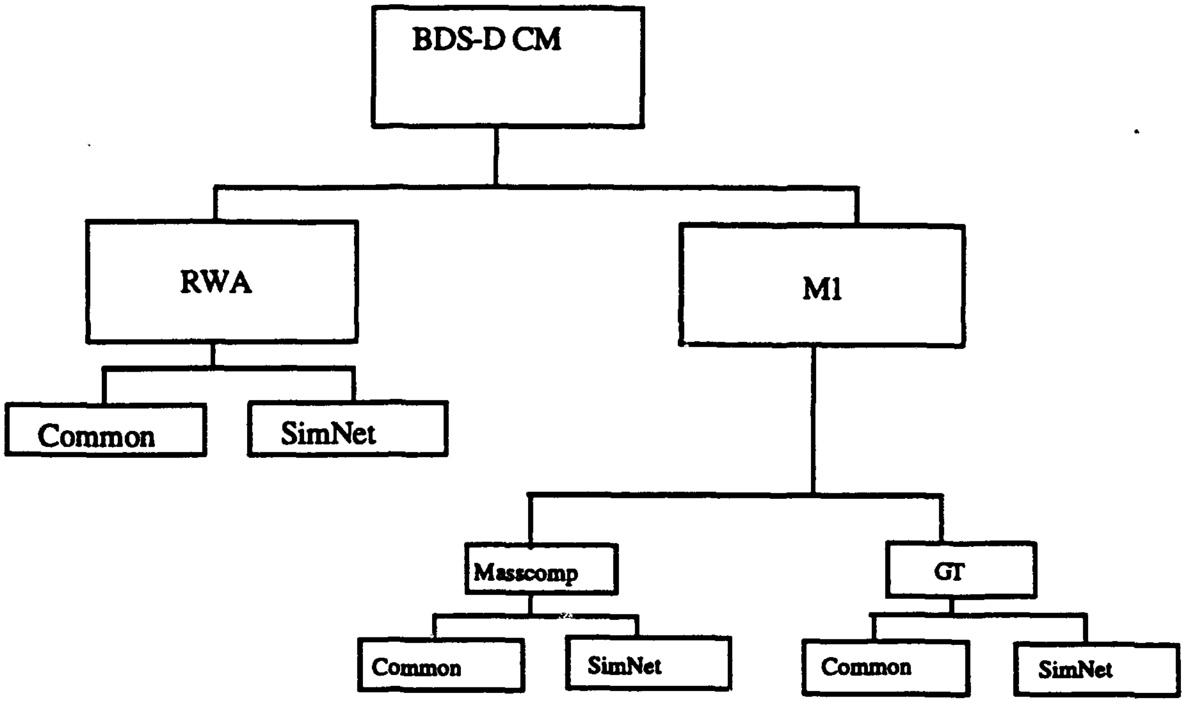


Figure 4-5. SSL Directory Structure

4.2.6 CM Tools/Build Environment

A collection of tools, as described below, have been installed for code control and building of software files. In addition, tools using UNIX script files have been prepared to automate and streamline the process of code control.

4.2.6.1 Revision Control System (RCS)/Concurrent Version System (CVS)

Revision Control System (RCS) is a program librarian for the tracking and control of baselined software in the UNIX Environment. This on-line system maintains baselined files in project libraries. RCS keeps the original versions of library files and then tracks the changes to the project library by storing the changes made with each retrieval and replacement file in the library. As a result, RCS can reconstruct any previous version of a project file. In addition to storing successive changes to library files, RCS monitors library access and keeps a historical record of library access. By entering RCS commands, the CM Staff can easily retrieve information about library transactions and contents. It provides CM and program management with an excellent status accounting tocl for version descriptions, and annotated file listings. This tool will be used to tag classes of a build to aid in the configuration identification and movement of code to build areas and maintenance of baseline areas. The Concurrent Version System (CVS) is a layered product on top of RCS. The CVS tool provides automation of repetitive RCS tasks and facilitates in the implementation of software identification and control.

4.2.6.2 Version Master System

The Version Master System is a program librarian for the tracking and control of baselined software in the Macintosh Environment and provides similar functionality as the RCS system described above.

4.2.6.3 MAKE

UNIX Operating System along with RCS provides an automated build tool called MAKE. MAKE automates and simplifies the building of software systems. In the development of a large software system many of the dependent source files are typically in various states of completion. When changes are made to the software system, MAKE determines which source files need to be recompiled, and ensures the software system is recompiled and linked with all the latest changes. MAKE interacts with RCS and extracts files from the RCS libraries when building new object files.

5.0 Interface Management

Loral's Program Engineering Manager (PEM) will be responsible for integrating all functions for the engineering aspects of the contract. The PEM, supported by the Process Control Supervisor will be responsible for ensuring that stated CM tasks are accomplished to program standards, on time, and within budget. The Process Control Supervisor will also be responsible to the Program Manager for assessing and reporting the cost impact of CM requirements on the actions of all other functions in the program during the course of the contract.

5.1 Documentation

The configuration management process must support a fast and accurate decision-making process for proposed engineering changes. Baseline documentation must be maintained throughout the development and operational process.

Currently, ADST assets are only partly documented. Documents may not reflect changes or revisions made to the ADST equipment, and no formal, coordinated process has been used to review and approve changes. The minimal set of documentation must be identified by the configuration control board and supplemental documentation for existing SIMNET-D equipment, procedures, and data must be brought under configuration management.

Supplemental documentation is the equipment documentation and manuals prepared by the ADST systems engineering organization based on their assessment of the current configuration and status of all government furnished ADST equipment documentation and manuals. Supplementary documentation is required for two purposes: (1) to support the maintenance and enhancement of the system, and (2) to facilitate user interconnection with the system and re-use of system modules. For each purpose each element of the system has potentially a hierarchy of documentation detail starting with the element's objectives and proceeding to its functional description, interface descriptions, user manuals, and maintenance manuals.

Through the preparation of supplemental documentation, the system will be described to sufficient detail to support maintenance and to support a general level of foreseeable enhancements. An appraisal process to be completed within one year, will determine the appropriate level of documentation for each system element.

New system elements will be added through DOs for experiments and system enhancements. The documentation needed to maintain and integrate the new system elements associated with each such DO will be developed as a part of that DO. In that way the overall ADST documentation will be kept up-to-date.

In some cases, a new DO will require documentation of existing system elements to a higher level of detail than included in the overall task of preparing supplemental documentation as part of the LSE. In those cases, that additional baseline-system documentation would also be included in the effort associated with the DO.

5.2 Interface Control

LORAL WDL shall operate the system interface control program through the CCB. Systems Engineering and analysis will be performed to ensure optimum and compatible hardware/software, hardware/hardware, and software/software interfaces. These interfaces will be documented and controlled by Interface Specifications and Interface Control Drawings as specified and funded for any particular Delivery Order system.

6.0 Configuration Traceability

6.1 Nomenclature

The process of nomenclature assignment and the requirements for titling is accomplished through the ADST Orlando technical library. Procedures for this process are documented in the LORAl Documentation Plan for the ADST Program.

6.2 Documentation Numbering

A numbering system is used for program CDRL's, and specifications. These numbers are assigned by the Orlando technical library. Procedures for this process are documented in the LORAl Documentation Plan for the ADST Program.

6.3 Hardware Identification

A comprehensive list of all hardware identified in the Loral Software Development Facility is currently being maintained by the LORAL PMO. The ADST System Definition Document also provides a description of theses systems.

6.4 Software Identification

Software is identified in Version Descriptions documents, the SSL, and in the CM Build Load plans. Refer to Sections 3 and 7 within this plan.

7.0 Configuration Status Accounting

Configuration Status Accounting will be the means through which control and tracking of discrepancies and change requests affecting CI/CSCIs are reported to STRICOM, and engineering managers concerned.

The Configuration Status Accounting contains a Configuration Identification List, DO Status Log, and Version Descriptions Documents. Whenever a status change occurs, the Configuration Identifications Data Base is updated to reflect its current status. Approved DOs to the product baseline developmental configurations are listed, thus providing a status accounting of DOs against each CSCI.

This data is kept and maintained in the 4D SP/CR data base server. This data base will be capable of providing complete configuration identification and status of each Delivery Order.

7.1 Data Bank Description

7.2 Data Bank Content

The SWCCP does not support a formal Data Bank, however, Configuration Management will control documents in both electronic and paper form. Systems Requirements Specification, System Design Specifications, Source Code, I&T Plan, Test Procedures, Test Reports, and other Documents required by the DO's will be controlled on-line or in file cabinets, book shelves, notebooks, etc. A tape or disk backup library will be maintained under Configuration Control for all software packages

7.3 Reporting

Weekly reviews with Configuration Management, Process Control, SW Engineering Management, and SQA are held with the PEM and PMO. Reporting activity status will be as follows:

- Weekly activity progress
- Monthly WAD/WP reviews
- Weekly milestone reviews
- Periodic Build Plan reviews
- Inputs for PMO/Customer Briefs

8.0 Configuration Management Audits

Formal configuration audits and reviews for the ADST program are not a requirement of this contract. However, informal technical reviews and technical audits :will be part of the surveillance throughout the life cycle of the project.

In addition, a Software Quality Assurance (SQA) function will be in place and is responsible for monitoring internal configuration management procedures that ensure compliance with the SWCCP and CMP. The SQA representative provides a continuing check on all software maintained for the ADST program. SQA coordinates and periodically audits the configuration maintenance activities to ensure consistency between the controlled documentation, software, and database elements comprising each DO configuration. SQA works with CM to assure that baseline, status accounting, library and change control procedures are being appropriately presented to the CCB. SQA will conduct a review of all products in conjunction with a major software baseline capture and/or release, in lieu of a PCA/FCA.

9.0 Subcontractor/Vendor Control

Within the context of the Configuration Management Plan the rules that apply to the Loral engineering staff will also apply to subcontractors. In particular, turnover of software developed on the ADST contract and its associated Delivery Orders will be required including cold start procedures, data tables, and documentation as specified in the Statement of Work. Any proprietary rights must be identified prior to issuing the subcontract and mutually agreeable between Loral, the subcontractor, and the government.

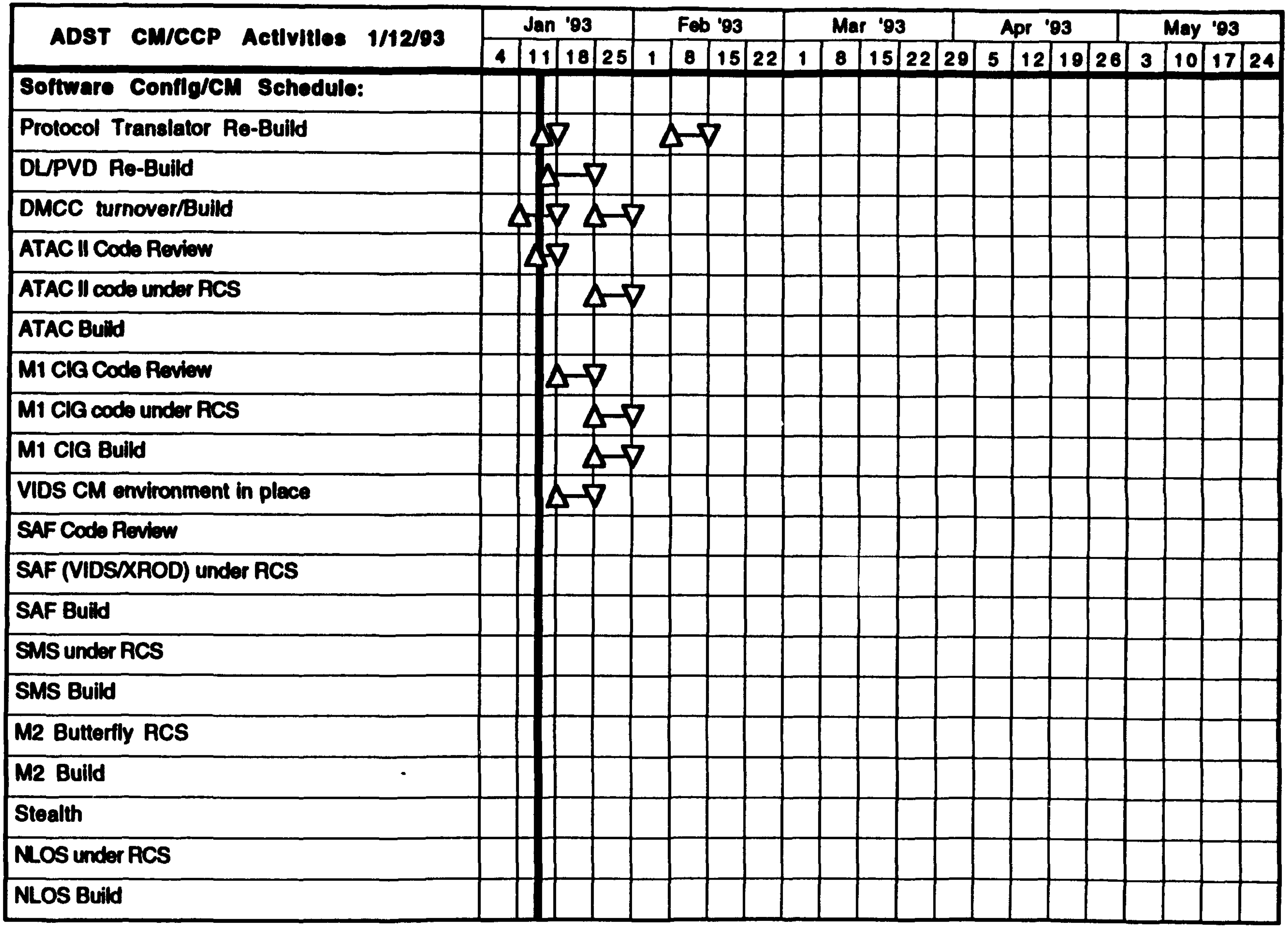
The software provided by the subcontractor must be built under CM control and issued to an Integration and Test team to "selloff" the software. This is advantageous for several reasons. First, it ensures that all code is under CM control. Second, it guarantees that what is being demonstrated and tested is traceable to the source code under control. Finally, liens that are levied against a build under CM control can be worked off in an orderly manner, and allow sufficient regression testing to take place, ensuring that a "fix" does not break other functionality. Such traceability and accountability provide a reusable baseline for incremental improvements to the next delivery order.

APPENDIX A. ADST TOP LEVEL PROGRAM SCHEDULE

Appendix A depicts the current program schedule at the time of publication of this document.

APPENDIX B. CM MILESTONE SCHEDULE

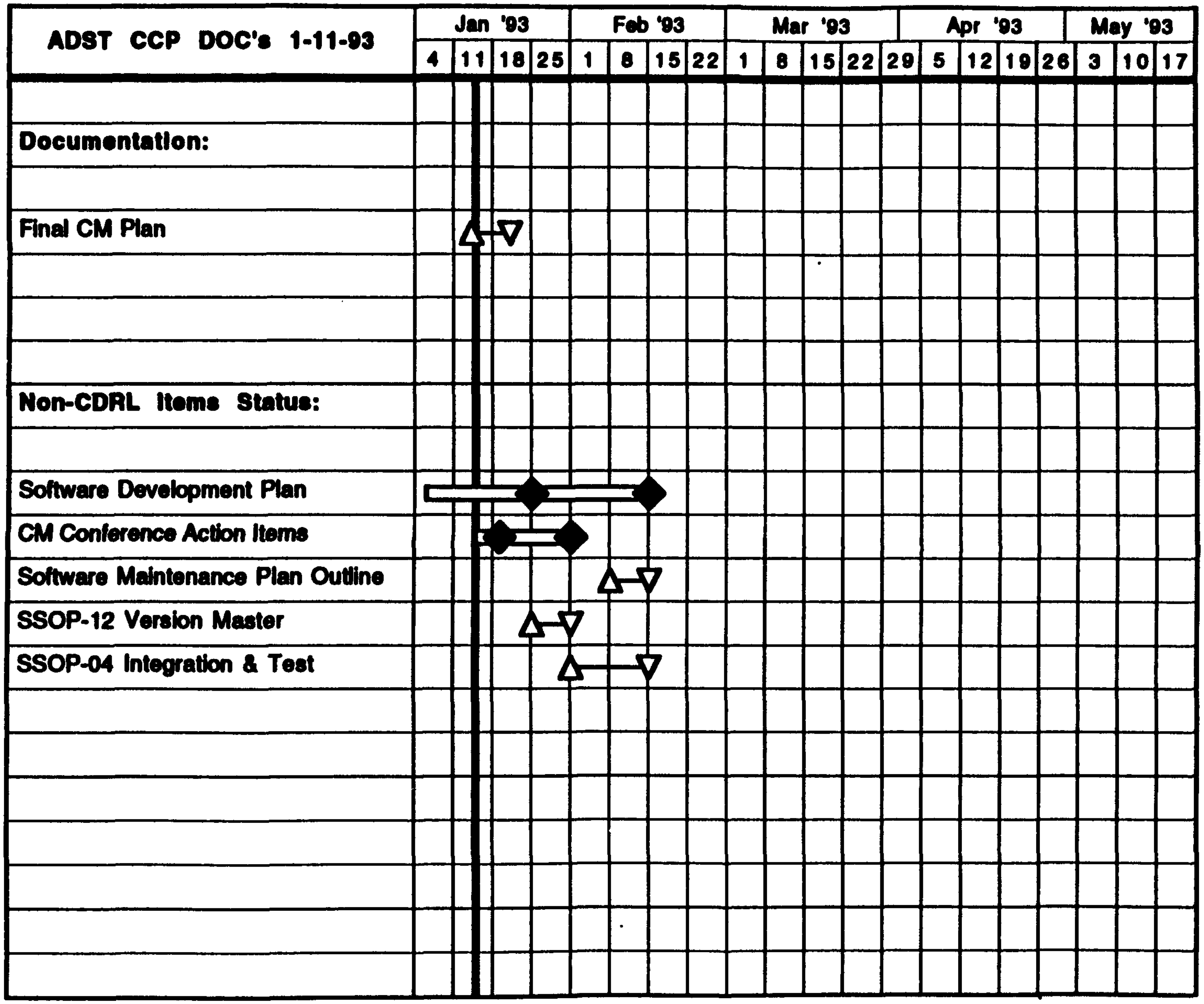
Appendix B depicts the current CM Activity schedule at the time of publication of this document.



ADST CM/CCP Activities 1/12/93	Jan '93				Feb '93				Mar '93					Apr '93				May '93			
	4	11	18	25	1	8	15	22	1	8	15	22	29	5	12	19	26	3	10	17	24
Software Config/CM Schedule:																					
Protocol Translator Re-Build		△	▽			△	▽														
DL/PVD Re-Build		△		▽																	
DMCC turnover/Build		△	▽	△	▽																
ATAC II Code Review		△	▽																		
ATAC II code under RCS				△	▽																
ATAC Build																					
M1 CIG Code Review			△	▽																	
M1 CIG code under RCS				△	▽																
M1 CIG Build				△	▽																
VIDS CM environment in place			△	▽																	
SAF Code Review																					
SAF (VIDS/XROD) under RCS																					
SAF Build																					
SMS under RCS																					
SMS Build																					
M2 Butterfly RCS																					
M2 Build																					
Stealth																					
NLOS under RCS																					
NLOS Build																					

ADST CM/CCP Activities 1/12/93	Jan '93				Feb '93				Mar '93					Apr '93				May '93			
	4	11	18	25	1	8	15	22	1	8	15	22	29	5	12	19	26	3	10	17	24
DI under RCS																					
DI Built																					
Data Base Utilities/tools RCS																					
Data Base Build																					
FWA under RCS																					
FWA Build																					
M1 Masscomp under RCS																					
M1 Masscomp Build																					
CVCC under RCS																					
CVCC Build																					

ADST CCP DOC's 1-11-93	Jan '93				Feb '93				Mar '93					Apr '93				May '93		
	4	11	18	25	1	8	15	22	1	8	15	22	29	5	12	19	26	3	10	17
VDD and COLDSTART																				
BDS-D RWA 1.0.0 VDD/CS	Completed Nov 92																			
BDS-D M1 1.0.0 VDD/CS		△	▽																	
BDS-D M1/XROD 1.1.0 VDD/CS		△	▽																	
BDS-D MCC 1.0.0 VDD/CS		△	▽																	
BDS-D DMCC 1.0.0 VDD/CS								△	▽											
BDS-D PT 1.0.0 VDD/CS								△	▽											
BDS-D DL 1.0.0 VDD/CS								△	▽											
BDS-D M1/CIG 1.0.0 VDD/CS																				
BDS-D M2 1.0.0 VDD/CS																				
BDS-D NLOS 1.0.0 VDD/CS																				
BDS-D Stealth 1.0.0 VDD/CS																				
BDS-D SAF 1.0.0 VDD/CS																				
BDS-D DI 1.0.0 VDD/CS																				
BDS-D ATACII 1.0.0 VDD/CS																				
BDS-D FWA 1.0.0 VDD/CS																				
BDS-D Database Utilities VDD/VS																				
BDS-D CVCC 1.0.0 VDD/CS																				
VIDS VDD/CS																				


ADST CCP DOC's 1-11-93
Jan '93
Feb '93
Mar '93
Apr '93
May '93
4
11
18
25
1
8
15
22
1
8
15
22
29
5
12
19
26
3
10
17
Documentation:
Final CM Plan
Non-CDRL Items Status:
Software Development Plan
CM Conference Action Items
Software Maintenance Plan Outline
SSOP-12 Version Master
SSOP-04 Integration & Test

APPENDIX C. SOFTWARE STANDARD OPERATING PROCEDURES (SSOP)

Appendix C defines the current set of SSOP's as referenced in the Configuration Management Plan. These SSOP's will be updated and additional SSOP's prepared, as required, per SSOP-001 procedures.

SSOP NO.	TITLE
SSOP-0001	SSOP PURPOSE/PROCEDURE
SSOP-0002	CONFIGURATION CONTROL BOARD (CCB) CHARTER
SSOP-0003	PROBLEM REPORTING PROCEDURE
SSOP-0004 (In Process)	INTEGRATION AND TEST PROCEDURE
SSOP-0005	CODE CHECKOUT/FTP
SSOP-0006 (In Process)	PROBLEM REPORTING ACCOUNTING PROCEDURE
SSOP-0007	CM TURNOVER PROCEDURE
SSOP-0008	RCS HEADER COMMENT STANDARDS
SSOP-0009	DELIVERY ORDER RELEASE KITS
SSOP-0010	CM/SDF BUILD LOAD DISTRIBUTIONS
SSOP-0011	REVISION CONTROL SYSTEM (RCS)
SSOP-0012	VERSION MASTER SYSTEM
SSOP-0013	CHANGE CONTROL/BUILD PROCEDURE FIELD SUPPORT
SSOP-0014	SP/CR AUDIT TRAILS

ADST
SOFTWARE STANDARD OPERATING PROCEDURE

TITLE: SSOP PURPOSE/PROCEDURE SSOP #0001

DATE: 9/28/92

AUTHOR: K. HUMBER MOD:

CCB PROCESS CONTROL
APPROVAL:____________________DATE:____________ APPROVAL__________DATE:______

PURPOSE: The Advanced Distributed Simulation Technology (ADST) Software Standard Operating Procedure (SSOP) serves as the procedural instruction set for all baseline maintenance elements. The SSOP master file will be maintained by the Software Configuration Manager and each member of the engineering staff shall be required to hold, use, and keep current an individual copy of the SSOP.

Any member of the engineering maintenance staff may submit data or redlines for inclusion in the SSOP. All new or redlined data shall be submitted to the Software Configuration Manager as an enclosure to a Software Problem/Change Report (SP/CR). The SP/CR will then be submitted to the Software Configuration Review Board (SCRB) for review and approval prior to its inclusion in the SSOP.

SSOP inputs shall address all aspects of baseline maintenance to include:

Procedures for handling software changes
Procedures for handling documentation changes
Instructions for using and completing baseline maintenance related forms
Procedures for maintaining the SSOP
Procedures for the SCRB
Procedures for software configuration management
Procedures for formal testing
Procedures for preparation of site deliveries
Procedures for interfacing with site engineering personnel

All SSOP inputs shall be prepared in the format provided in this document. This format is available from the Software Configuration Manager's Public Folder and utilizes commercially available Microsoft Word on the Macintosh computer.

SSOP numbers shall be assigned by the SCRB. All new inputs shall be submitted to the Software Configuration Manager with the SSOP number field blank and shall not be numbered until approved by the SCRB.

ADST
SOFTWARE STANDARD OPERATING PROCEDURE

TITLE: CONFIGURATION CONTROL BOARD CHARTER SSOP # 0002

DATE: 10/3/92

AUTHOR: K. Humber MOD:

CCB APPROVAL:________________ DATE:__________ PROCESS CONTROL APPROVAL________ DATE:_____

Charter: ADST LORAL WDL CONFIGURATION CONTROL BOARD (CCB)

The ADST LORAL WDL CCB responsibilities are to assess, review, and evaluate the necessity for proposed changes to approved product baselines. Release of new design(s), and changes to approved baselines require approval of the Loral WDL CCB. The CCB is chaired by the Loral ADST Program Manager and is composed of senior members of the various support organizations including Software Configuration Management. The CCB will meet on a as needed basis, the frequency and time of meetings to be determined by the chairperson.

The CCB will mediate conflicts between sites due to changes (hardware, firmware, documentation, and software product configuration items); categorize and prioritize changes as they are requested and approved; and analyze proposed DO updates to a software product baseline. Release of specifications, new design(s), and changes to approved baselines require approval of the CCB.

The primary reporting mechanism for changes to the BDS-D/DO system configurations is via the Software Problem/Change Report. The SP/CR provides the means for reporting errors in development, test, and operational software and/or hardware. An SP/CR is completed by any user or developer who discovers a software error, or, desires a modification in order to improve system operational efficiency. The SP/CR provides a formal means for recording all software and hardware faults and modification requests. Each SP/CR will be submitted to the CCB for analysis. SP/CR's which are out of scope, such as beyond the normal schedule, scope of current activities, budget or enhancements/improvements will be forwarded to the LORAL WDL Program office for resolution.

Data submitted to the CCB shall include, but not be limited to, the following:

- Performance problems with the software baseline
- Operational discrepancies with the software baseline
- Proposed or directed changes to the software baseline
- Proposed or directed changes to identified hardware
- New or revised drawings
- New or revised Software Support Operating Procedures
- New or revised build procedures
- New or revised command files
- New or revised data formats
- New or revised delivery instructions
- New or revised documentation updates
- New or revised software configuration management policies/procedures
- Any information which will enhance team performance or baseline maintenance

The ADST CCB shall be composed of the following:

CCB Chairman

The CCB chairman shall preside over all CCB meetings. In the absence of the CCB chairman, the vice chairman shall preside. The current CCB chairman is the ADST Program Manager, Jerry Novak

CCB Vice Chairman

The CCB vice chairman will coordinate and preside over all CCB meetings when the chairman is absent. The current CCB vice chairman is Rick Bright.

Customer Representative

The customer representative will act as the liaison between BDS-D Sites, and the LORAL WDL for all actions addressed at the CCB. The current customer representative is the BDS-D Manager, Jim Exter.

CCB Recorder

The Software Configuration Manager shall serve as the permanent CCB recorder. In the absence of the CCB recorder, an alternate recorder shall be designated. The current CCB recorder is Maria Ipsaro.

CCB Orlando Representative

The Site Coordinator in Orlando shall serve as the liaison between the Orlando Program Office and the WDL CCB. The current Orlando site representative is Paul Hinote.

Site CCB Representatives

The site CCB representatives are Randy Kubik from Ft. Rucker and Jimmy Adams from Ft. Knox.

CCB Permanent Members

All Engineering Technical Leads, Primary Test Engineer, and the program SQA representative are permanent members of the ADST CCB. Attendance for permanent members is mandatory at any scheduled CCB.

CCB Member At-Large

CCB membership shall include all engineers assigned to the ADST maintenance staff. Attendance by non-permanent members shall be based upon the agenda and/or the need for outside technical support.

All data submitted to the CCB shall be in the form of a Software Problem/Change Request (SP/CR) form. SP/CR's written at the sites will be reported to the designated site coordinator and then to the CCB.

The basic flow of SP/CR data through the CCB is as follows:

Receipt of SP/CR

All incoming SP/CRs are entered into the SP/CR data base by the submittor. An E-mail message is sent to the Software Configuration Manager/CCB Recorder, so the SP/CR is added to the agenda for the next CCB.

CCB Review of New SP/CR

All new SP/CRs will initially be assigned to a responsible engineer for analysis. The engineer shall prepare a written proposed solution/analysis which provides information concerning the scope and complexity of the required change. Once the proposed solution is accepted by the CCB, the assignee can begin the fix.

Receipt of CCB Direction

Upon receipt of CCB direction, the SP/CR shall be placed in the status designated by the CCB. Refer to the CCB DISPOSITION INSTRUCTIONS of this SSOP for information concerning actions to be taken in response to CCB direction.

The following defines disposition instructions in response to actions assigned by either the CCB:

IMPLEMENT

Responsible engineer will be assigned to implement the approved workplan. The engineer shall prepare required code/changes in the development environment, shall conduct informal unit testing to ensure the change is error free, and shall redline or draft required documentation changes associated with the change. Upon completion of all required actions, the responsible engineer shall submit to the CCB listings of all new/modified code, copies of all new/redlined documentation, listings of all new/modified command files, utilities, build procedures, or other files associated with the formal baseline. Specific due dates will be assigned for implementation.

DEFER

SP/CRs may be assigned to deferred status if the CCB determines that the proposed change is inappropriate for implementation at this time. Deferred SP/CRs are not reviewed at weekly CCB meetings but may be reopened at a future time. Due dates for deferred SP/CRs shall be INDEF.

BASELINE

The Software Configuration Manager shall update the appropriate software or documentation baseline with the changes submitted by the responsible engineer. Upon completion of the baselining process, the Software Configuration Manager shall prepare an updated build of all affected software and shall notify the CCB that the software is available for informal system test. Specific due dates shall be assigned to SP/CRs assigned for baselining.

TEST

The independent software system test engineer shall prepare test procedures and conduct required system testing of the modified software baseline. Additionally, the software system test engineer shall perform applicable regression testing to ensure the change has not affected other software functionality. Upon completion of informal system testing, the software system test engineer shall inform the CCB that the software has been successfully tested and shall submit formal test procedures to the CCB, if applicable. Specific due dates shall be assigned to SP/CRs assigned for test.

CLOSE

For SP/CRs, closure shall be authorized upon completion of all implementation and test activities and acceptance by the CCB.

WITHDRAWN

This status shall be assigned to SP/CRs which the originator has elected to remove from the formal processing cycle. In most cases, SP/CRs will only be withdrawn when it has been determined that no problem actually exists, or that there is a solution to this problem available within the current baseline.

ANALYSIS

The basic actions to be taken during analysis are defined on the pages of this SSOP titled CCB SP/CR PROCESSING. It should be noted that major system enhancements may result in a workplan which recommends generation of a formal ECP for submission to the ADST program office. In most cases, such recommendations shall be limited to changes or enhancements which are of such size or complexity that they cannot be completed with the current release schedule or costing.

The CCB shall assign priorities to all SP/CRs. The priority assigned is used to convey the urgency associated with the change. Authors should recommend a priority when drafting SP/CRs. Responsible engineers should ensure higher priority SP/CRs are worked before lower priority SP/CRs.

IMMEDIATE

This priority is reserved for problems which result in one of the following conditions:

Entire system is rendered inoperable
Major capability is rendered inoperable
System is experiencing major periods of downtime
System is unable to perform basic operational capability

The CCB recorder shall call an immediate CCB to disposition the SP/CR and correction is expected to occur within 24-48 hours. If at all possible, coordination with site personnel shall be handled via phone and/or fax. Once the immediate problem has been solved, the SP/CR shall be reassigned priority URGENT.

URGENT

This priority is reserved for SP/CRs which are recommended or directed for implementation as part of the next scheduled software release. SP/CRs in this priority reflect problems which impact daily operations but which do not require immediate correction. In most cases, such SP/CRs will address issues such as minor processing discrepancies which can be accommodated with workarounds, minor enhancements to existing capabilities, or discrepancies which can wait for the next scheduled release. SP/CRs in this priority will normally be resolved as quickly as possible and will be closed upon completion of required actions.

ROUTINE

This priority is reserved for SP/CRs which involve recommended or direct changes which have minimal operational impact. In this case, implementation is desired, but can wait for a release which can accommodate lower level priority problems.

ENHANCEMENT

This priority is reserved for recommended enhancements which are not essential to operational system performance. Such enhancements will only be implemented when directed by the ADST Program Office.

ADST
SOFTWARE STANDARD OPERATING PROCEDURE

TITLE: PROBLEM REPORTING PROCEDURE SSOP # 0003

DATE: 10/3/92

AUTHOR: K. Humber MOD:

CCB
APPROVAL:________________DATE:__________

PROCESS CONTROL
APPROVAL________DATE:_____

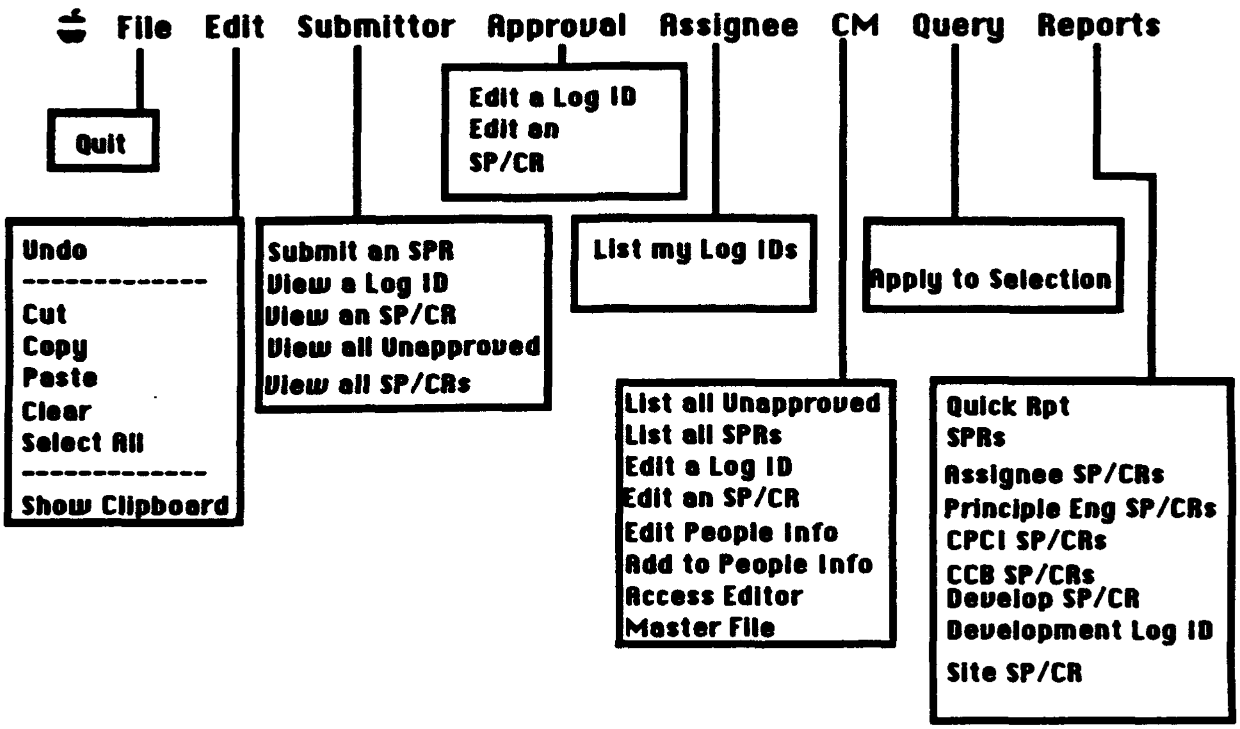
This procedure addresses the SP/CR form and provides detailed instructions concerning completion of all applicable data. The ADST SP/CR Database System is a multi-user Software Problem Reporting application using the 4th Dimension DBMS. It was designed to address the need for a low cost Macintosh based SP/CR system.

A Quick & Dirty User's Manual is attached for your reference in the use of the tool and the SP/CR screens. The manual includes field name and type descriptions, pull-down menu selections and explanations. Also attached is a copy of the SP/CR front page form.

User's are placed into one of four functional groups: Administration, Approver, Assignee, and CM. These groups have distinct menu bars which define the limit of that group's access to the data within the SP/CR system and parallels the functions to be performed by the group.

The diagram below depicts all SP/CR menus and menu items; however, the exact menus appearing during an SP/CR session depend on the user's group type. Refer to the Quick & Dirty User's Manual for details on the functions of each type of user.

ADST SP/CR



For further information, please see Maria Ipsaro or Pete Peterson from the ADST Configuration Management team. They will have "Administrator" accounts in order to add new accounts and maintain existing accounts.

ADST
SOFTWARE SUPPORT OPERATING PROCEDURE

TITLE: CODE CHECK IN/OUT USING FTP SSOP 0005

DATE: 20 October 1992

AUTHOR: M.E. Ipsaro MOD:

CCB APPROVAL:________________DATE:__________ PROCESS CONTROL APPROVAL________DATE:____

PURPOSE: Identify a procedure for checking code in/out from the CM environment using FTP. These steps must be followed to ensure integrity of the CM baseline and traceablity of the software.

The most current version of each file resides on the CM disk in the respective directory tree. Read-only access is granted to all files under CM control. Prior to making any modifications and in order to obtain the most correct version of a file, all files must first be copied from the CM environment into an engineer's development directory using "cp" or "FTP". Revision Control System (RCS) is the code management tool used by CM to store and maintain all released versions of files and baselines.

I. COPYING THE CURRENT VERSION OF A FILE LOCALLY

1. Login into a SUN using your account name and password.
2. Set your default to your working directory.
3. Enter the command to copy the designated file(s):

```
cp /a3/adst-cm/XXX/filename(s) .
```

II. COPYING MODIFIED FILE(S) TO THE CM TURNOVER DIRECTORY USING FTP

1. Log into FTP using the account name and password.
2. Set your default directory to the location of the modified file(s) for turnover.
3. Enter the command to copy the modified file(s) on to the CM disk:

```
put filename(s) /a3/turnover/XXX
```

3. Once the file(s) have been copied to the turnover directory logout of FTP.
4. Login using TELNET and set default to the turnover directory and reset file privileges. This permits the owner to have read, write, and execute privileges, with group and world having no privileges.

```
cd /a3/turnover/XXX
chmod 700 filename(s)
```

5. Send a mail message to CM notifying them of code turnover.

ADST
SOFTWARE SUPPORT OPERATING PROCEDURE

TITLE: CM Turnover Procedure SSOP 0007

DATE: 20 October 1992

AUTHOR: M.E. Ipsaro MOD:

CCB APPROVAL:________________DATE:__________ PROCESS CONTROL APPROVAL________DATE:____

PURPOSE: Identify a formal method of turning modified code over for incorporation into the CM baseline. This procedure creates an organized approach for transitioning modified code from an engineer's workspace to a temporary location for capture and placement into the controlled baseline by CM.

Prior to code turnover, the responsible engineer must perform a successful software build in the their development environment. The engineer must also be able to load and demo the application on the target machine (if necessary).

For initial turnover of code, the engineer must turnover over all source code required to build the application (.c, .h, makefiles, .txt, .def, etc.,) and run the application (.d, .p, .com, etc.,),. Test plans, test scripts and test data is an integral part of the initial turnover. If this data exists at initial turnover or if it is created subsequently, it is essential that it is turned over to CM for control. Engineers are not required to turnover any extraneous files such as object files and libraries, .old, .save, and executable files. These file types will not be accepted as part of the initial turnover. CM reserves the right to refuse the turnover if it contains extraneous (garbage) files. CM will insert the required Revision Control System (RCS) header information upon initial turnover for the ".c", ".h", and "makefiles". At this time, RCS header information is not required on any other file type. The engineer must follow the Prologue & RCS Header Comment Block Standard set forth in SSOP 0008 for new files and modified files.

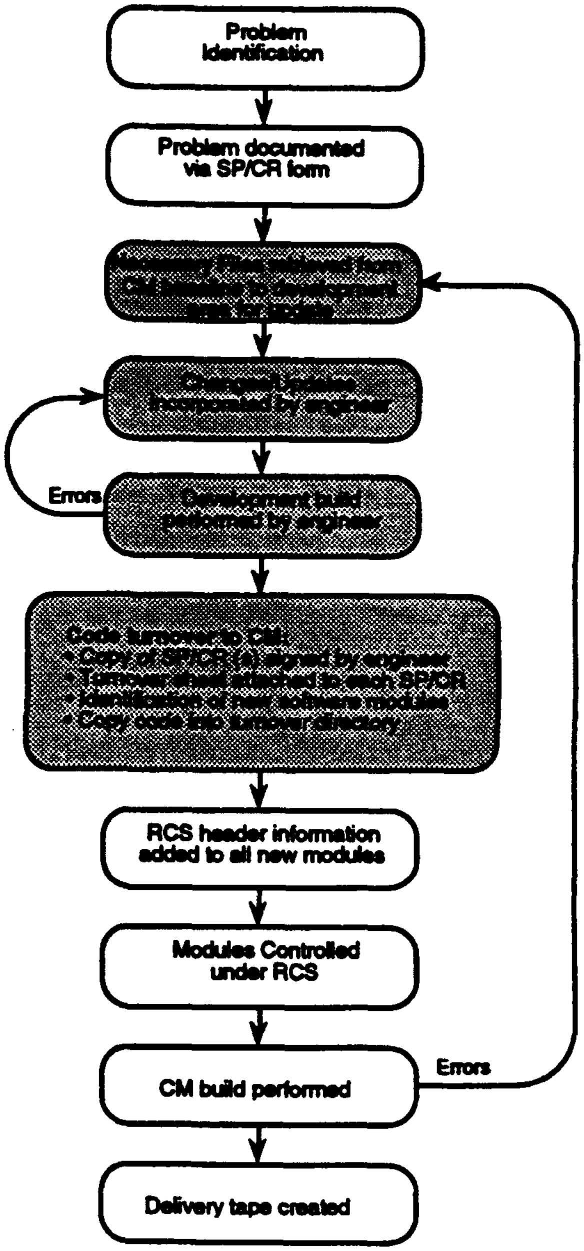
Software build and installation procedures are required to be turned over with the initial turnover of code. Updates to these procedures will be turned over by the responsible engineer as necessary. The build procedures must identify the build steps required and the executable file(s) that will be produced as a result of a successful compilation and link.

Code is ready for turnover upon successful completion a development build and unit testing of all changes. A turnover directory is identified for each DO. This directory is open for read, write, and delete access by engineers. The responsible engineer will copy the modified source code and/or application files into the designated turnover location on the CM disk. (For example, the RWA code is copied to /a3/turnover/RWA). The responsible engineer must notify CM requesting a build of the initial turnover or the latest changes to the baseline. For each turnover, the engineer must prepare and deliver a list of files being turned over to CM. (A sample turnover sheet is attached to this SSOP). If the list of files is extensive, a directory listing may substitute the turnover form as long as the directory path is provided for each file, the SP/CR number and the DO name are specified, and new files identified. This turnover list documents what is being turned over for the build and serves as a self check for the engineer and CM. The turnover directory will be copied to the CM environment for incorporation into RCS and the contents of the turnover directory deleted.

For incremental turnovers (subsequent builds after initial turnover), engineers must generate a Software Problem/Change Report (SP/CR) if a file modification is required. (Refer to SSOP 0003 for specific instructions on the use of SP/CRs). The modified file(s) must be documented with the appropriate SP/CR information in the header comment block and audit trailed in the body of the code. (Refer to SSOP 0014 for information on audit trails).

The latest controlled version of the CM baseline is located on /a3/adst-cm/directory. If a modification to a file is required, engineers must obtain the latest version from the CM baseline. Files on the CM disk are open for read-only access and can be copied to an engineer's workspace for modification. The location of the CM baseline for each DO is in the /a3/adst-cm directory tree. If an engineer wishes to review and/or modify a previous version of a file(s) that is under CM control, the engineer must make a request from CM. The CM engineer will check out a read-only copy of the previous version of a file(s) and copy it to the engineer's workspace. (This procedure is only temporary until a development environment is established for each DO. A copy of the CM environment will be created for software engineering. It will contain the entire directory tree and the RCS subdirectories. This is the engineering development environment. Engineers will manage and control their development changes. At anytime in the development process they can checkout, review and/or modify previous versions of a file(s)). All changes will be submitted back to the central Software Support Library (SSL) baseline.

ADST SP/CR & Turnover Flow


Problem Identification
Problem documented via SP/CR form
Necessary Files retrieved from CM baseline to development area for update
Changes/ Updates incorporated by engineer
Errors
Development build performed by engineer
Code turnover to CM:
• Copy of SP/CR(s) signed by engineer
• Turnover sheet attached to each SP/CR
• Identification of new software modules
• Copy code into turnover directory
RCS header information added to all new modules
Modules Controlled under RCS
CM build performed
Errors
Delivery tape created

▨ Shaded area corresponds to SSOP 0007 CM Turnover Procedures

ADST SOFTWARE TURNOVER FORM

[redacted]	LORAL □ SITE □ ____________ Site Name	DATE:
ORIGINATOR:	Ext.	Discipline:
Delivery Order (DO):	Build Version:	
List of updated files		

Directory Path Name:	Filename:	√ New
Signature Date		

ADST SOFTWARE TURNOVER FORM
ver. 0.0 12/2/92

ADST
SOFTWARE STANDARD OPERATING PROCEDURE

TITLE: Prologue & RCS Header Comment Block Standard SSOP 0008

DATE: 1 October 1992

AUTHOR: M. E. Ipsaro MOD: 15 January 1993

CCB APPROVAL:__________________DATE:__________ PROCESS CONTROL APPROVAL________DATE:____

PURPOSE: The purpose of this SSOP is to provide guidance and standards to produce source file prologues; and guidance and standards to prepare your file for Revision Control System (RCS) control.

PROLOGUE:

Source lines shall be no longer than 78 characters.

Each source file must contain a prologue in the following format:

```
/********************************************************
* File Name:
* System:
* Project:
* Description:
*
*     Version     Date     Author     Title              SP/CR No.
*
*
********************************************************/
/********************************************************
*
* Target Platform:
* Special Hardware:
* OS & Version:
* Compiler & Version:
* Makefile/Project:
* Compiler Options:
*
********************************************************/
```

```
/************************************************************
* Reference:
*
* Description:
*
* Inputs:
*
* Outputs:
*
************************************************************/
/************************************************************
*
* SP/CR No.   Description of Modification
*
*
************************************************************/
```

RCS HEADERS:

The version control tool that will be utilized on the program for tracking code modifications is Revision Control System (RCS). RCS requires that header information be inserted before the prologue at the top of each ".c", ".h", and Makefile . An example of each is listed below.

Example of RCS header information for a .c file:

```
/* $Header$ */
/*
 * $Log$
*/
static char RCS_ID[] = "$Headers$";
```

Example of RCS header information for a .h file:

```
/* $Header$ */
/*
 * $Log$
*/
```

Example of RCS header information for a Makefile:

```
# $Header$
# $Log$
```

ADST
SOFTWARE STANDARD OPERATING PROCEDURE

TITLE: Delivery Order Release Kit SSOP #0009

DATE: 26 October 1992

AUTHOR: K. L. Humber MOD: 2 December 1992

CCB APPROVAL:________________DATE:__________ PROCESS CONTROL APPROVAL________DATE:_____

PURPOSE: This SSOP outlines the elements required to support a Delivery Order Software release as required by the sponsoring organization. The elements required for a Delivery Order package may vary depending on the release requirements. Each kit will be available for general release unless specified by the customer for limited release.

Each Release Kit will be comprised of the following List of items:

1. Delivery of a Version Description Document depicting the entire software baseline capture (or delta from the previous baseline build), to include new, deleted, and updated modules and directory structure. The format for the VDD will adhere to DI-MCCR-80013A, CDRL requirement for the SW Support CCP. The VDD will be will be generated on the MAC in Microsoft WORD and distributed to the sites on a floppy disk along with one hardcopy. The VDD will include, but not limited to:

   a. System Overview
   b. Inventory of Materials Release
   c. Inventory of CSCI contents:
      provide a snapshot of the RCS (current version) source libraries at the time the build "freeze" is created. Review RCS libraries/structures for adherence to CM practices, policies, and procedures.
   d. Changes Installed:
      identification of SP/CR's that are associated with this release (open, closed, pending).
   e. Build and Distribution Instructions.
   f. Regression Test Results. Provide tools to support any necessary on-site regression testing, if applicable
   g. Interface Compatibility
   h. Adaptation data:
      Unique to site data, if applicable.
   j. Summary of changes:
      provide softcopy SP/CR's and minutes from the CCB for review
   k. Release Notes:
      installation instructions. Depict notes/differences/compatibility since the last release.
   l. Identification of known problems and/or discrepancies.
   m. Identification of the hardware configuration required by a software release.

n. Reference to the Acceptance Test Plan for use to recertify a configuration has been restored to its "baseline" configuration. (Recertification may be accomplished by rerunning an acceptance test).

2. Coldstart Procedures. A written list of instructions that allow construction of executable images based on source code installed on a system support configuration. This includes procedures required to move and install the compiled/linked executable image(s) to a target configuration. The procedures include startup, shutdown, operational modes (as required), and installation and distribution instructions, interaction with other simulators and hardware compatibility notes (as applicable).

   The Coldstart Procedures will contain the following statement regarding the certification of the software as being free from security threats to the best of our knowledge: "I (name of CM Manager) on this (date) hereby certify that the software release (name of software release) has been built from a limited access, controlled baseline. This software is, to the best of my knowledge, free of malicious code intended to subvert its operation."

3. Identification of command files required for the build configuration setup, and distribution process. Identification of data files, data bases, etc., required to bring up the system.

4. English narrative synopsis of functionality of the released system.

5. Documentation updates, including drawing configuration changes, as applicable.

6. Standard header information for ease of tape identification, to include the following information:

   (1) Project Title and Government Contract Number/Date
   (2) Title of software program
   (3) Date and release number of software
   (4) Sponsoring Agency
   (5) Security Classification
   (6) Name, address (office symbol) and phone number of agency who can release software for use by another agency/contractor
   (7) Name and phone number of contractor POC for release

Some Delivery Orders (i.e., VIDS) also require the following release kit documentation:

1. Software Maintenance Manual (similar to the VDD format). This manual will focus on describing details of how to perform a build, change control, and description of supporting documentation. The elements that comprise the Software Maintenance Manual are:

   a. Delivery Order Overview (architectural, software and hardware, interfaces)

   b. Hardware/Software Descriptions

   c. CM cold start procedures, distribution and installation procedures of a build.

   d. CM change control procedures and reporting of problems.

e. Documentation Cross References (functional description, test plan, operator's/user's manual)

2. Operator's/User's Manual comprised of the following:

a. Operations Concept

b. Operations (startup, shutdown, operational modes, interaction with other simulators).

ADST
SOFTWARE STANDARD OPERATING PROCEDURE

TITLE: CM/SDF BUILD LOAD PROCEDURES **SSOP # 0010**

DATE: 10/3/92

AUTHOR: B. MOHNING MOD:

CCB APPROVAL:________________ DATE:__________ PROCESS CONTROL APPROVAL________ DATE:_____

PURPOSE: The Configuration Management-Software Development Facility Build Load Distributions Procedure is the instruction set for downloading builds from the CM environment to the development environment.

1. Back up any files on the target machine that you may overwrite. A list of files to be installed will be kept in the directory containing the build.

2. From the CM account, change directory to the build you want to download. All build packages are stored on /a3/BUILD/“DO Name” (i.e. /a3/BUILD/RWA). The files you need will be stored in tar format and called “DO name”.“version”.tar. FTP down to the target machine. Do this from the command prompt by typing ftp “target machine name” and pressing return. FTP will then prompt you for your password on that machine. After you have entered your password, you will get the ftp> prompt. Now change directory to where you want to load the tarfile. Once there, type put “package name” and hit return. Your tar file will now be loaded on the target machine. When the ftp> prompt comes back, type bye and press return. This will take you out of FTP and return you to your local machine.

3. Login on the target machine. You can do this by using either rlogin or telnet. To use rlogin, type rlogin “machinename” and press return. To use telnet, type telnet “machinename” and press return. In either case, you will prompted for a password, enter your password and hit return. When you get logged on, cd to the directory that contains the tar file you just downloaded in step 2. Once there, type tar xvf “tarfilename” and press return. You are now ready to run the new installation.

ADST
SOFTWARE SUPPORT OPERATING PROCEDURE

TITLE: Revision Control System SSOP 0011

DATE: 1 December 1992

AUTHOR: M.E. Ipsaro MOD:

CCB APPROVAL:________________ DATE:__________ PROCESS CONTROL APPROVAL________ DATE:_____

PURPOSE: Identify a version control tool which tracks many versions and configurations of the software system during program development and maintenance. Revision Control System (RCS) ensures that each software configuration is controlled in a manner which preserves accuracy and maintains up-to-date information. RCS is a software tool which manages revisions to source files and provides a method for storing and tracking changes to those files. It automates the storing, retrieval, identification and traceability of revisions, and it provides selection capabilities for composing software configurations.

All baselined files are located in RCS subdirectories for each directory in the directory tree. All source files required to build the application, run the application, test scripts, test data, and tools are contained in RCS. Files turned over by the engineer for control are checked into RCS by the CM engineer. The CM engineer ensures that the appropriate RCS header information (see SSOP 0008) is contained in all the "*.c", "*.h", and "make" files before placed into RCS. All the files will then be checked into RCS using the check in command with an appropriate message denoting the description and/or reason for the checkin and/or modification. A read-only copy will be retained in the directory tree. For example;

```
ci -u  -m'Initial Turnover' *.*
```

The checkout command extracts the latest version from RCS and places it in the working directory. The file can be edited, and, when finished, checked back into RCS. This command checks out a file from RCS with a lock to prevent accidental overwrites. For example;

```
co  -l -m'Initial Turnover' filename
```

A previous version can be checked out for read only, by using the- *r* option. For example;

```
co -r1.1 -u  -m'work in progress' filename
```

A complete history of changes is accumulated as log messages that are requested at check in. To view the history of changes to a RCS subdirectory enter the following command:

```
rlog RCS/*
```

To view the history of changes to a particular file, enter the following command:

rlog filename

The rcsdiff command compares revisions of a files contained in RCS. The following command is used to run differences between revision 1.1 and 1.2:

rcsdiff -r1.1 -r1.2 filename

The rcsfreeze command records a software configuration. Rcsfreeze assigns a symbolic number to a given revision in all RCS files thus taking a snapshot of all the files at a fixed point in time.

Listed in this SSOP are a few examples of the RCS commands that are used to checkin, checkout, view a history of changes to files, and run differences. For more information on RCS and its options, refer to the document ***RCS - A System for Version Control*** by Walter F. Tichy, dated 3 January 1991 and the man pages on the UNIX system.

ADST
SOFTWARE SUPPORT OPERATING PROCEDURE

TITLE: **Version Master System** **SSOP # 0012**

DATE: 14 January 1993

AUTHOR: M. Ipsaro MOD:

CCB APPROVAL:__________DATE:______ PROCESS CONTROL APPROVAL______DATE:____

PURPOSE: To describe the Version Master tool used on the Macintosh for controlling and managing any type of Macintosh files.

Version Master is a document version management and control product for the Apple Macintosh. Version Master manages the creation and modification of files on a Macintosh. It stores all versions of a file in a single icon, and it allows a user to annotate each new version providing traceability to when and why certain changes were made. Version Master stores documents in a central database within a Project folder. Documents are transferred to and from ("checked in" and "checked out") of the Version Master database by using the Version Master DA and by using the Version Master extension to the standard open dialog of most applications.

Version Master helps users coordinate changes when working together on a product or working alone. It provides information about document changes such as: who has modified a particular document, what changes have been made, who is currently modifying the document, and the complete revision history. Users are able to work with the latest versions of documents and can easily integrate changes with those of other users. Earlier versions are archived for easy retrieval.

An important feature of Version Master is document modification interlocking. A user cannot modify a document if another user is modifying the same document. This is especially important in a multi-user environment where many users share the same documents.

ADST
SOFTWARE STANDARD OPERATING PROCEDURE

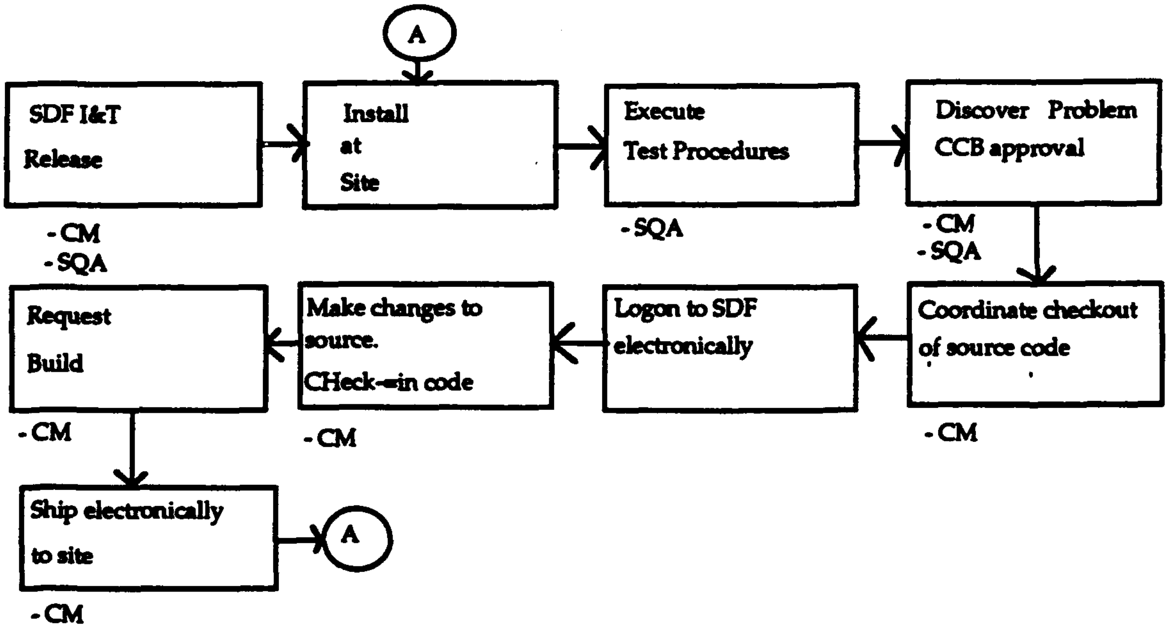
TITLE: CHANGE CONTROL/BUILD PROCEDURE FIELD SUPPORT SSOP # 0013

DATE: 10/16/92

AUTHOR: K. HUMBER MOD:

CCB APPROVAL:________________DATE:__________ PROCESS CONTROL APPROVAL________DATE:____

PURPOSE: This procedure and diagram depict the process for ADST Site Software Engineers who require the use of the LORAL WDL Software Development Facility (SDF) for debugging and building software. In the event the site development configuration does not provide the correct environment, an engineer can perform their tasks using the equipment in the SDF by logging into the EIEN network, allowing them to log on remotely, make changes, initiate a build, and transfer it over the link. The coordination of this process will be in cooperation with the LORAL WDL CM staff and the LORAL WDL SQA staff. All known problems during I&T will be documented on the Problem Reporting Form as described in SSOP-0003, and processed through the Configuration Control Board.


A
SDF I&T Release
-CM
-SQA
Install at Site
Execute Test Procedures
-SQA
Discover Problem CCB approval
-CM
-SQA
Coordinate checkout of source code
-CM
Logon to SDF electronically
Make changes to source. CHeck-=in code
-CM
Request Build
-CM
Ship electronically to site
-CM
A

ADST
SOFTWARE SUPPORT OPERATING PROCEDURE

TITLE: SP/CR Audit Trails SSOP 0014

DATE: 2 December 1992

AUTHOR: M.E. Ipsaro MOD:

CCB APPROVAL:__________ DATE:_______ PROCESS CONTROL APPROVAL_______ DATE:____

PURPOSE: Identify a standard for incorporating Software Problem/Change Report (SP/CR) audit trails when source code is added, deleted or modified. A SP/CR audit trail history which documents the description and reason for the change is required in the header comment block for each change made to a new or modified file. Lines of code are not required to be documented with a SP/CR audit trail within new modules only in the header comment block. SP/CR audit trails provide traceability of each change made to a module during the software development and maintenance phases of a project.

The following is a list of SP/CR audit trail standards:

a. The prologue header comment block contains the audit trail history data. The most recent SP/CR audit trail history shall be documented as the last entry in the header comment block.

b. The SP/CR audit trail shall be entered as a comment at the end of the modified line of code. For example:

```
/* SP/CR  100* /
```

c. Multiple SP/CR audit trails affecting the same line(s) of source code shall be concatenated. For example:

```
/*  SP/CR2,SP/CR36,SP/CR100  */
```

d. New modules added in response to a SP/CR shall have an initial audit trail entry in the header comment block which identifies the module as new and associates it with the SP/CR. Subsequent lines of code within the new module need not be annotated with the SP/CR audit trail.

e. When code is added to a module, insert the SP/CR audit trail and following comments in the beginning and end of the new lines of code. This shall be in addition to the normal audit trail history in the header comment block at the beginning of the module.

```
/*  SP/CR155  START OF NEW CODE ADDED */
```

insert new code

```
/* SP/CR155 END OF NEW CODE ADDED */
```

f. When code is deleted, a SP/CR audit trail and following comments shall be documented at the point of code deletion. The code shall not be physically deleted. Each line of deleted code should be annotated with a comment symbol. (Be careful that the comment symbol is not embedded in what you are deleting because this will cause an error during compilation). This shall be in addition to the normal audit trail history provided in the header comment block at the beginning of the module.

```
/* SP/CR155 END OF CODE DELETION*/
/* add comment symbol before and after*/
/* each line of deleted code */
/* SP/CR155 END OF CODE DELETION*/
```

g. The date which is used to document an SP/CR audit trail entry in the header comment block shall take the format DD-MMM-YYYY. For example:

30-Nov-1992

APPENDIX D. GLOSSARY

Acceptance Testing Formal testing conducted to determine whether or not a system satisfies its acceptance criteria.

Approval Formal recognition of the validity and acceptability of an action or a product.

Baseline A specification or product that has been reviewed and agreed upon, and thereafter serves as a basis for further development. A baseline can be changed only through change control procedures. (2) A configuration identification document or set of documents formally designated by the Government and fixed at a specific time during the system's life cycle. Baselines, plus approved changes from these baselines, constitute the current configuration.

Build The compilation and linking of source code into the program executable images that run on the target simulator. Included as part of the build are any data files or data bases required to initialize and/or support the execution of the programs in the simulators.

Build Load The program executable images that run on the target simulator and any data files or data bases required to initialize or support the operation of the simulator.

Configuration Control Board Focal point for all changes to the BDS-D software product baselines. The CCB will ensure proper application of configuration control and establish the basis for status accounting. The CCB will provide active program support for monitoring, coordinating, and implementing change control.

Configuration Identification The process controlling the approved products in a system and recording their characteristics, such as numbers and other identifiers affixed to the items and documents. The approved documents that identify and define the item's functional and physical characteristics in the form of specifications, drawings, associated lists, interface documents, and documents referenced therein.

Configuration Item An aggregation of hardware, firmware, software, or any of its discrete portions, which satisfies an end use function and is designated for configuration management.

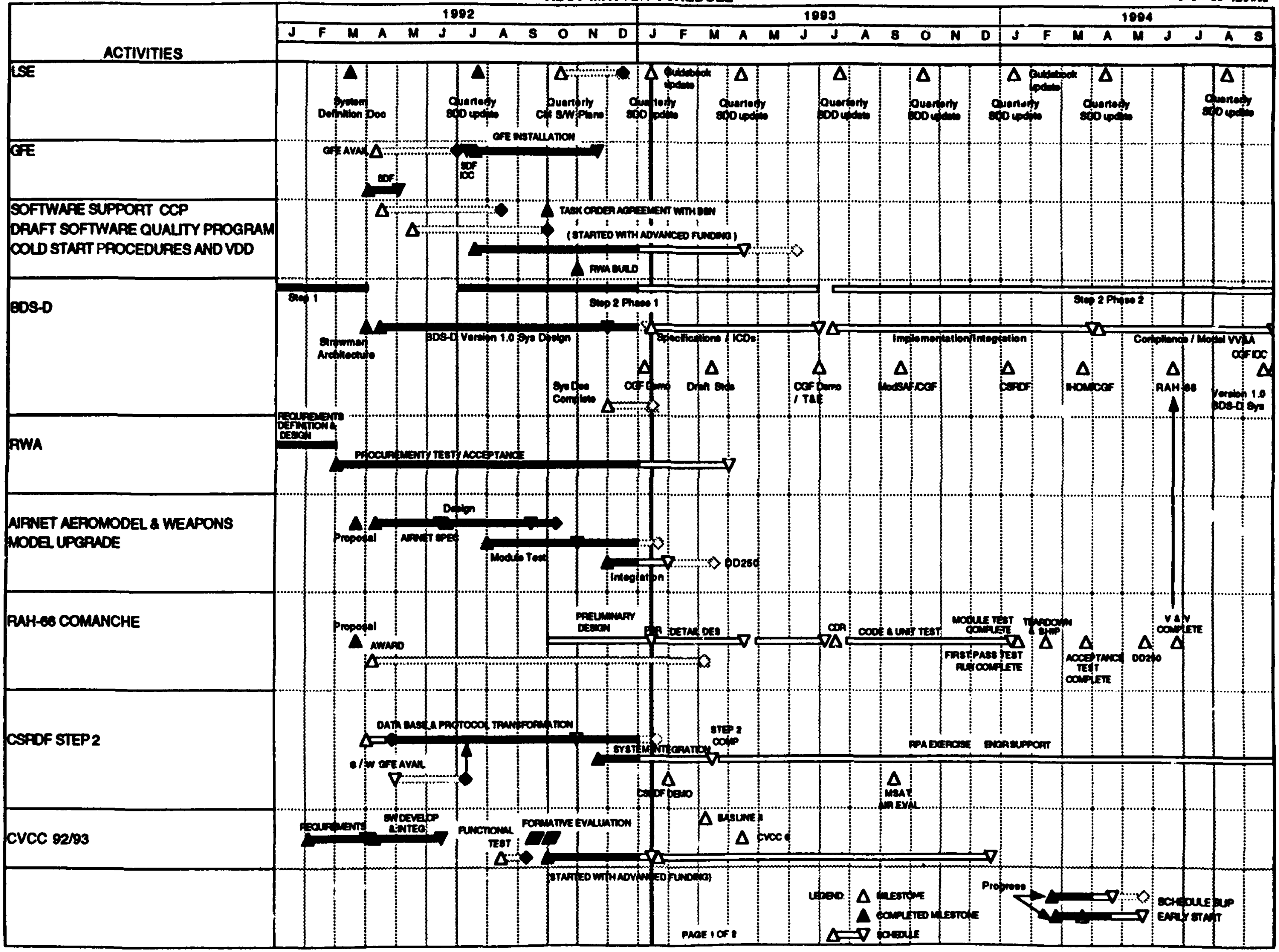
Configuration Management A discipline applying technical and administrative direction and surveillance to (a) identify and document the functional and physical characteristics of CIs, (b) control changes to CIs and related documentation, (c) record and report change information needed to manage CIs effectively, including the

APPENDIX E. ACRONYMS

ADST	Advanced Distributed Simulation & Training
BDS-D	Battlefield Distributed Simulation-Development
CCB	Configuration Control Board
CM	Configuration Management
CMP	Configuration Management Plan
CI	Configuration Item
CSCI	Computer Software Configuration Item
DBMS	Data Base Management System
DO	Delivery Order
ECP	Engineering Change Proposal
EDMS	Engineering Data Management System
EDS	Electronic Defense Systems
FCA	Functional Configuration Audit
FCI	Functional Configuration Item
GFE	Government Furnished Equipment
H/W	Hardware
IAW	In Accordance With
IDD	Interface Design Document
IRS	Interface Requirements Specification
I&T	Integration and Test
PCA	Physical Configuration Audit
PEM	Program Engineering Manager
POC	Point of Contact
PMO	Program Management Office
RCS	Revision Control System
SCM	Software Configuration Management
SDD	Software Design Document
SDF	Software Development File
SDP	Software Development Plan
SIMNET	Simulation Networking
SMM	Software Maintenance Manual
SOW	Statement of Work
SPC	Software Process Control
SP/CR	Software Problem/Change Report
SPS	Software Product Specification
SQA	Software Quality Assurance
SRS	Software Requirements Specification
SSL	Software Support Library
SSOP	Software Support Operating Procedures
STD	Software Test Description
STP	Software Test Plan
S/W	Software
SWCCP	Software Support Contract Change Proposal
TRR	Test Readiness Review
VDD	Version Description Document
V&V	Verification and Validation
WDL	Western Development Labs

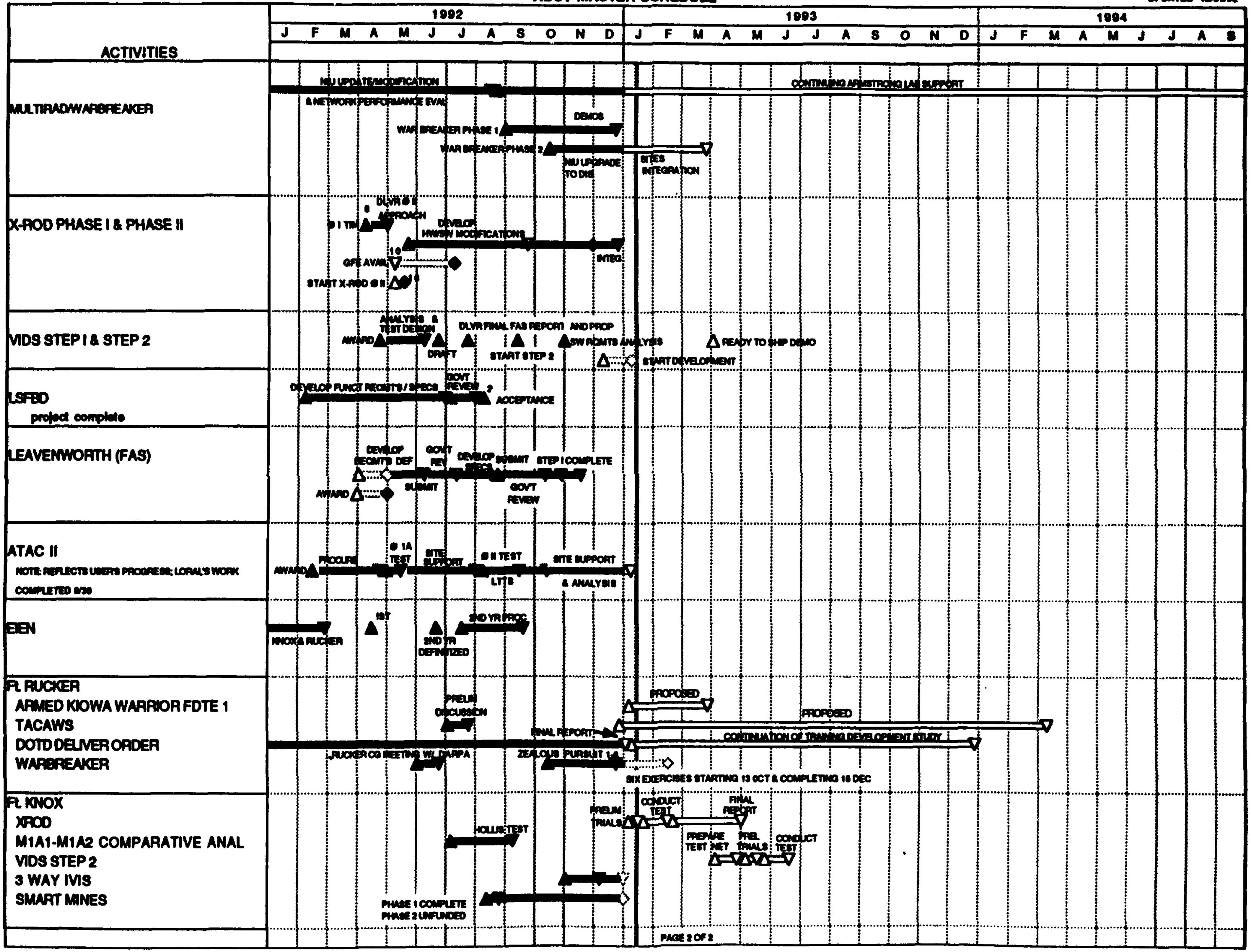
ADST MASTER SCHEDULE

UPDATED 12/31/92



ADST MASTER SCHEDULE

UPDATED 12/31/92


ACTIVITIES
J F M A M J J A S O N D J F M A M J J A S O N D J F M A M J J A S
MULTIRAD/WARBREAKER
NIU UPDATE/MODIFICATION
& NETWORK PERFORMANCE EVAL
CONTINUING ARMSTRONG LAB SUPPORT
WAR BREAKER PHASE 1
DEMOS
WAR BREAKER PHASE 2
NIU UPGRADE TO DIS
SITES INTEGRATION
X-ROD PHASE I & PHASE II
@ I TIM
DLVR @ II APPROACH
DEVELOP HW/SW MODIFICATIONS
GFE AVAIL
INTEG
START X-ROD @ II
VIDS STEP I & STEP 2
AWARD
ANALYSIS & TEST DESIGN
DRAFT
DLVR FINAL FAS REPORT AND PROP
START STEP 2
SW RQMTS ANALYSIS
START DEVELOPMENT
READY TO SHIP DEMO
LSFBD
project complete
DEVELOP FUNCT REQMT'S / SPECS
GOVT REVIEW
ACCEPTANCE
LEAVENWORTH (FAS)
DEVELOP REQMTS DEF
AWARD
GOVT REV
SUBMIT
DEVELOP SPECS
SUBMIT
GOVT REVIEW
STEP I COMPLETE
ATAC II
NOTE: REFLECTS USER'S PROGRESS; LORAL'S WORK COMPLETED 9/30
AWARD
PROCURE
@ 1A TEST
SITE SUPPORT
@ II TEST
LTTS
SITE SUPPORT
& ANALYSIS
EIEN
KNOX & RUCKER
1ST
2ND YR DEFINITIZED
2ND YR PROC
Ft. RUCKER
ARMED KIOWA WARRIOR FDTE 1
TACAWS
DOTD DELIVER ORDER
WARBREAKER
PRELIM DISCUSSION
PROPOSED
PROPOSED
FINAL REPORT
CONTINUATION OF TRAINING DEVELOPMENT STUDY
RUCKER CG MEETING W/ DARPA
ZEALOUS PURSUIT 1-4
SIX EXERCISES STARTING 13 OCT & COMPLETING 18 DEC
Ft. KNOX
XROD
M1A1-M1A2 COMPARATIVE ANAL
VIDS STEP 2
3 WAY IVIS
SMART MINES
HOLLIS TEST
PRELIM TRIALS
CONDUCT TEST
FINAL REPORT
PREPARE TEST NET
PREL TRIALS
CONDUCT TEST
PHASE 1 COMPLETE
PHASE 2 UNFUNDED